



Inês Margarida Santos Gaspar

Licenciada em Ciências da Engenharia Electrotécnica e de Computadores

Simulador de Elevador do Tipo Monta-Cargas

Dissertação para obtenção do Grau de Mestre em
Engenharia Electrotécnica e de Computadores

Orientador: Luís Filipe Figueira Brito Palma, Professor Auxiliar,
Faculdade de Ciências e Tecnologia da Universidade
Nova de Lisboa

Júri

Presidente: Doutor Luís Filipe dos Santos Gomes, FCT-UNL
Arguente: Doutor João Almeida das Rosas, FCT-UNL
Vogal: Doutor Luís Filipe Figueira Brito Palma, FCT-UNL



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Março, 2020

Simulador de Elevador do Tipo Monta-Cargas

Copyright © Inês Margarida Santos Gaspar, Faculdade de Ciências e Tecnologia, Universidade NOVA de Lisboa.

A Faculdade de Ciências e Tecnologia e a Universidade NOVA de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objetivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

AGRADECIMENTOS

Sendo esta a secção mais pessoal deste trabalho, gostaria de agradecer a todas as pessoas que, direta ou indiretamente, contribuíram para a finalização desta etapa.

Em primeiro lugar, um grande obrigado ao meu orientador, Luís Brito Palma, por toda a orientação e disponibilidade ao longo deste trabalho.

Agradeço também aos meus amigos e colegas de curso pela sua amizade, que preencheu esta etapa da minha vida com ótimos momentos, repletos de companheirismo, força e humor, agradecendo em particular à Débora, Mara, Mariana, Matilde, Miguel, Sofia e Rita, pela amizade, pelo apoio e compreensão, por todas as gargalhadas e diversão que me proporcionaram no decorrer de todo o curso.

Ao Diogo, agradeço todo o apoio e compreensão, especialmente ao longo do desenvolvimento deste trabalho e por todos os bons momentos que passámos no decorrer do curso.

Por fim, mas sempre em primeiro lugar no meu coração, agradeço à minha família.

Muito obrigado por terem sempre acreditado em mim.

Ao meu irmão, Afonso, o meu agradecimento por todo o apoio e disposição.

Aos melhores pais do mundo, agradeço toda a espera por este momento e todos os esforços feitos para a minha formação ao longo destes anos. Muito Obrigado!!

Do fundo do coração, um grande obrigado a todos!!!

"A 5 year degree attained after 7 years is still a degree.

A graduation at the age of 50 is still a graduation.

A Mercedes bought at the age of 65 is still a Mercedes.

Don't let people bully you with their timelines of success.

You got this."

RESUMO

O elevador é o meio de transporte mais comum para mover pessoas ou cargas entre pisos. Com o uso diário de elevadores em diversos ambientes, desde hospitais a edifícios residenciais, torna-se necessário otimizar e inovar os modelos existentes, melhorando a qualidade do serviço, desde a minimização do tempo de espera até à poupança de energia.

No setor industrial, os sistemas de controlo e automação são cada vez mais requisitados, recorrendo essencialmente a controladores lógicos programáveis (PLC). Deste modo, torna-se necessário testar cenários hipotéticos de processos e sistemas de forma a evitar situações perigosas e perda de produtividade. A única ferramenta que pode permitir melhorar e testar esses sistemas complexos e dinâmicos é a simulação.

O objetivo principal deste trabalho consiste em desenvolver um simulador de um elevador do tipo monta-cargas em linguagem *Processing*. Este simulador é baseado no protótipo de um elevador monta-cargas que se encontra no laboratório 2.3 do departamento de Engenharia Electrotécnica e de Computadores, na Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa.

Para o efeito foi necessário a construção da interface gráfica com o utilizador (GUI) e a implementação dos modos de funcionamento do elevador. O elevador monta-cargas pode ser controlado a partir de dois modos de funcionamento: modo automático e modo manual. No modo automático é escolhida uma sequência de funcionamento, estando esta sequência associada a um código em linguagem lista de instruções (IL). No modo manual, o elevador recebe ordens de um conjunto de botões que permitem seleccionar o piso pretendido (Piso 0, Piso 1 ou Piso 2).

No futuro, este simulador será utilizado nas unidades curriculares de automação e tecnologia de controlo, como recurso didático no ensino das linguagens de programação lista de instruções (IL), *Ladder* (LD) e diagrama funcional em sequência (SFC).

Palavras-chave: Simulador, Elevador, Monta-Cargas, Controlador Lógico Programável (PLC), Linguagem em Lista de Instruções (IL), Linguagem *Processing*.

ABSTRACT

The elevator is the most common way of transport for moving people or loads between floors. With the daily use of elevators in different environments, from hospitals to residential buildings, it is necessary to optimize and innovate the existing models, to improve the quality of service and minimize the waiting time to save energy.

In the industrial sector, control and automation systems are increasingly required, essentially using programmable logic controllers (PLC). Therefore, it is needed to test hypothetical scenarios of processes and systems in order to avoid serious damages and productivity loss. The only tool that can improve and test these complex and dynamic systems is simulation.

The main objective of this project is to develop a simulator of a freight elevator using the Processing language. This simulator is based on the prototype of a freight elevator, placed on the Department of Electrical and Computer Engineering, at the Faculty of Science and Technology of Universidade Nova de Lisboa.

For this purpose, it was necessary to build the graphical user interface (GUI) and the implementation of the elevator's operating modes. The freight elevator can be controlled from two operating modes: automatic mode and manual mode. In automatic mode, a sequence of operation is chosen. That sequence is associated with an instruction list code (IL). In manual mode, the elevator receives orders from a set of buttons that allows the selection of desired floor (Floor 0, Floor 1 or Floor 2).

In the future, this simulator will be used in automation and control technology areas as a teaching resource in teaching programming languages such as instruction list (IL), Ladder (LD) and sequential function chart (SFC).

Keywords: Simulator, Elevator, Freight Elevator, Programmable Logic Controllers (PLC), Instruction List Language (IL), Processing Language.

ÍNDICE

Lista de Figuras	xiii
Lista de Tabelas	xv
Siglas	xvii
1 Introdução	1
1.1 Contexto e Motivação	1
1.2 Objetivos	2
1.3 Contribuições	2
1.4 Estrutura do Documento	3
2 Estado da Arte	5
2.1 Introdução	6
2.1.1 Terminologia dos Elevadores	6
2.1.2 Legislação no Fabrico de Elevadores	8
2.2 Automação Industrial	9
2.2.1 Conceito de Simulação	9
2.2.2 Controladores Lógicos Programáveis	11
2.2.3 Programação de Autómatos	15
2.3 Linguagem <i>Processing</i>	16
2.3.1 História e Características Principais	16
2.3.2 Aplicações	17
3 Modelo Proposto, Tecnologias e Implementação	19
3.1 Desenvolvimento do Simulador	20
3.1.1 Metodologia Proposta	20
3.1.2 Tecnologias Utilizadas	21
3.2 Especificação do Simulador	28
3.2.1 Descrição do Funcionamento do Simulador	28
3.2.2 Autómata Finito aplicado ao Funcionamento do Simulador	29
3.2.3 Especificações funcionais	32
3.2.4 Arquitetura do Sistema	34

3.3	Implementação do Simulador	35
3.3.1	Mecanismos de Simulação	35
3.3.2	Visualização e Animação	39
3.3.3	Interface Gráfica com o Utilizador (GUI)	43
4	Resultados Experimentais	57
4.1	Modos de funcionamento do Simulador	58
4.1.1	Modo de Funcionamento Manual	58
4.1.2	Modo de Funcionamento Automático	61
4.1.3	Paragem de Emergência	66
4.2	Validação do Simulador	68
5	Conclusões e Trabalho Futuro	69
5.1	Conclusões	70
5.2	Trabalho Futuro	71
	Bibliografia	73
A	Apêndice	77
A.1	<i>Processing</i> - Função <i>Lerp()</i>	77
I	Anexos	79
I.1	Anexo 1 - Entradas e Saídas Disponíveis no PLC	79
I.2	Anexo 2 - Características PLC TSX Micro 3721	81
I.3	Anexo 3 - Disposição dos Sensores, dos Fins de Curso e dos Ímanes	84

LISTA DE FIGURAS

2.1	Exemplo de um PLC - Modelo Siemens S7-300.	13
2.2	Exemplos de PLCs compactos e modulares.	13
2.3	<i>Scan cycle</i> do PLC [22].	14
2.4	O <i>Processing</i> está interligado com várias linguagens de programação [26]. . .	16
2.5	Exemplo de como desenhar um retângulo [26].	17
2.6	Exemplos de trabalhos desenvolvidos em <i>Processing</i>	18
2.7	Exemplo de um jogo usando <i>Processing</i> e <i>Arduino</i> [12].	18
3.1	Protótipo de elevador do tipo monta-cargas.	21
3.2	Equipamentos da parte de controlo do elevador.	22
3.3	Sensores do protótipo.	24
3.4	Interface do simulador de elevador do tipo monta-cargas.	28
3.5	Funcionamento do simulador.	29
3.6	Exemplo de Autómata Finito do modelo do simulador.	30
3.7	Máquina de estados finitos do modo manual.	31
3.8	Máquina de estados finitos do modo automático.	31
3.9	Arquitetura da metodologia proposta.	34
3.10	Interface gráfica do simulador.	35
3.11	Resumo dos botões utilizados no simulador.	38
3.12	Reposta visual do sensor AF e dos fins de curso (FCI e FCS).	40
3.13	Reposta visual dos botões PE e OFF/ON.	41
3.14	Visualização do estado do simulador.	42
3.15	Interface gráfica com o utilizador (GUI) em linguagem <i>Processing</i>	43
3.16	Modo manual na interface gráfica do simulador. Elevador parado no piso 1. .	44
3.17	Modo automático na interface gráfica do simulador.	45
3.18	Modo automático na interface gráfica, quando é seleccionada a sequência 2. .	45
3.19	Grafcet - Botão de paragem de emergência (PE).	48
4.1	Estado inicial da interface gráfica do simulador.	58
4.2	Interface gráfica do simulador na transição Piso 0 » Piso 1.	59
4.3	Modo manual - Elevador parado no Piso 1.	59
4.4	Modo manual - Elevador parado no Piso 2.	60
4.5	Chegada da cabina ao piso 2.	60

4.6	Interface gráfica do simulador na transição da sequência 1 - Piso 0 » Piso 1. .	61
4.7	Interface gráfica do simulador na transição da sequência 1 - Piso 1 » Piso 0. .	62
4.8	Interface gráfica do simulador na transição da sequência 2 - Piso 0 » Piso 2. .	63
4.9	Cabina pronta para começar a descer, na transição da sequência 3 - P2 » P1. .	64
4.10	Interface gráfica do simulador na sequência 3 - Elevador parado no Piso 1. .	64
4.11	Interface gráfica do simulador no final da sequência 3, na transição P1 » P0.	65
4.12	A paragem de emergência (PE) é acionada.	66
4.13	Após ter sido acionado o botão de paragem de emergência, o elevador desce para o piso 0.	67
A.1	Função <i>Lerp()</i> - Movimento do Elevador.	77
I.1	PLC TSX Micro 3721 [14, 35].	81
I.2	PLC base e extensão e localização do módulo DMZ 28DR [14, 35].	82
I.3	Ligações do PLC TSX Micro 3721 [35].	83
I.4	Disposição dos sensores, dos fins de curso e dos ímanes ao longo das guias [35].	85

LISTA DE TABELAS

3.1	Lista de endereços das entradas/saídas utilizadas no simulador.	47
I.1	Lista de endereços de ligação das entradas/saídas do PLC TSX Micro 3721 .	80

SIGLAS

AF Ampola da Frente.

AI *Analog Input.*

AO *Analog Output.*

AT Ampola de Trás.

B0 Botão Seleção Piso 0 ou de Rearme (*Reset*).

B1 Botão Seleção Piso 1.

B2 Botão Seleção Piso 2.

BF Biestável da Frente.

BT Biestável de Trás.

CPU Unidade Central de Processamento (*Central Processing Unit*).

DI *Digital Input.*

DO *Digital Output.*

FA *Finite Automata.*

FBD Linguagem em Blocos de Funções (*Function Block Diagram*).

FCI Fim de Curso Inferior.

FCS Fim de Curso Superior.

GRAFCET Gráfico Funcional de Comando Etapa Transição (*Graphe Fonctionnel de Commande Étape Transition*).

GUI *Graphical User Interface.*

HMI Interface Homem-Máquina (*Human-Machine Interface*).

IL Linguagem em Lista de Instruções (*Instruction List*).

KM0 Relé de Descida.

KM1 Relé de Subida.

LD Linguagem em Diagrama de Contactos (*Ladder Diagram*).

MIT *Massachusetts Institute of Technolog.*

NA Contacto Normalmente Aberto.

NC *Normally Closed* (Contacto Normalmente Fechado).

NF Contacto Normalmente Fechado.

NO *Normally Open* (Contacto Normalmente Aberto).

P0 Piso 0.

P1 Piso 1.

P2 Piso 2.

PE Paragem de Emergência.

PLC Controlador Lógico Programável (*Programmable Logic Controller*).

SFC Diagrama Funcional em Sequência (*Sequential Function Chart*).

ST Linguagem Texto Estruturado (*Structured Text*).

VF Variador de Frequência.

INTRODUÇÃO

No presente capítulo são apresentados os principais objetivos e contribuições para o desenvolvimento do trabalho, bem como a estrutura da dissertação.

1.1 Contexto e Motivação

Atualmente a população urbana não consegue imaginar a sua vida sem elevadores. A forma como auxiliam no dia-a-dia torna-os numa ferramenta indispensável. Um grande número de modernos sistemas de elevadores são instalados na maioria dos edifícios mais altos, desde edifícios de habitação até aos edifícios industriais. Deste modo, torna-se necessário otimizar e inovar os modelos existentes, melhorando a qualidade do serviço, desde a minimização do tempo de espera até aos consumos de energia.

Um dos problemas na indústria é a necessidade em simular e testar previamente estes tipos de sistemas, mais precisamente os códigos implementados em autómatos, antes de estes serem aplicados aos modelos concebidos. Se existisse a possibilidade de simular e testar o funcionamento destes sistemas evitar-se-ia o risco de ocorrência de avarias neste tipo de máquinas.

No âmbito da tese de mestrado do aluno Nuno Ângelo foi desenvolvido um protótipo de um elevador monta-cargas. Este protótipo encontra-se no laboratório 2.3 do departamento de Engenharia Electrotécnica e de Computadores, na Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa.

1.2 Objetivos

O objetivo principal deste trabalho consiste no desenvolvimento de um simulador de um elevador do tipo monta-cargas, baseado no protótipo do elevador, em linguagem *Processing*.

Com o desenvolvimento do simulador vai ser possível estudar a linguagem lista de instruções (IL), aplicada ao movimento do elevador.

Para o desenvolvimento do simulador foram definidos os seguintes objetivos:

- **Compreensão do funcionamento do Elevador Monta-Cargas** – Definir quais as características a aplicar e como adaptá-las ao simulador;
- **Desenvolvimento do Simulador** – Criação da interface gráfica com o utilizador (GUI) do simulador em linguagem *Processing*;
- **Aplicação da Linguagem em Lista de Instruções (IL)** – Análise da linguagem em IL de forma a aplicar ao controlo do simulador;
- **Controlo do Simulador** – Controlo do simulador através de comandos manuais e automáticos, a partir de sequências de funcionamento associadas a instruções em código em linguagem IL.

1.3 Contribuições

Relativamente às contribuições deste trabalho, este contribuiu para:

- Desenvolvimento de um simulador, em linguagem *Processing*, de um elevador real, com elevado nível de aproximação, presente no laboratório 2.3 do departamento de Engenharia Electrotécnica e de Computadores;
- Este simulador permite estudar e compreender a linguagem de programação lista de instruções (IL), como também a tradução para as linguagens *Ladder* e *SFC*;
- O simulador oferece também a possibilidade de ser utilizado como recurso didático e pedagógico para o estudo e compreensão de modelos funcionais do tipo *Grafcet*.

1.4 Estrutura do Documento

A dissertação está organizada em cinco capítulos principais, estando estruturados da seguinte forma:

1. Introdução

O presente capítulo, o capítulo 1, trata o enquadramento do trabalho, os objetivos concretos para o desenvolvimento do trabalho e a estrutura do documento.

2. Estado da Arte

No capítulo 2 apresenta-se uma visão geral do enquadramento teórico do tema, introdução aos conceitos base sobre elevadores e uma resumida explicação das especificações das tecnologias utilizadas.

3. Modelo Proposto, Tecnologias e Implementação

O capítulo 3 descreve as características do modelo, a tecnologia envolvente e a implementação prática de acordo com os objetivos do trabalho.

4. Resultados Experimentais

O capítulo 4 é destinado à apresentação dos resultados obtidos, bem como a demonstração do funcionamento do simulador.

5. Conclusões e Trabalho Futuro

Por fim, no capítulo 5 apresentam-se as principais conclusões deste trabalho e sugerem-se possíveis desenvolvimentos que poderão ser feitos em trabalhos futuros.

O trabalho inclui, ainda, em anexo:

- **APÊNDICE 1:** *Processing* - Função *Lerp()*;
- **ANEXO 1:** Tabela dos endereços das entradas/saídas do PLC TSX Micro 3721;
- **ANEXO 2:** Características PLC TSX Micro 3721;
- **ANEXO 3:** Disposição dos sensores, dos fins de curso e dos ímanes ao longo das guias.

ESTADO DA ARTE

Este capítulo destina-se à apresentação teórica dos conceitos mais relevantes, enquadrados no tema da dissertação.

É feita uma breve introdução sobre a terminologia e legislação relativa aos elevadores.

Além disso, é abordado o conceito de simulação, controladores Lógicos programáveis (PLC), assim como as linguagens de programação de autómatos.

Neste capítulo é ainda abordada a linguagem *Processing*, nomeadamente as suas principais características e aplicações.

2.1 Introdução

Cidades altamente populosas em todo o mundo carecem de espaço imobiliário para muitos propósitos, desde edifícios residenciais a industriais, de tal forma que surgiu a necessidade de aumentar em altura os edifícios. Com a construção de edifícios cada vez mais altos foi necessário desenvolvimento de meios mecânicos que facilitassem a movimentação de pessoas e bens entre os vários pisos [3].

Estruturas de estacionamento de vários andares são uma área promissora a explorar, para atender a dificuldade em encontrar lugares de estacionamento. O uso eficiente do espaço numa estrutura de estacionamento requer ainda que as faixas de rodagem sejam eliminadas para que possam ser usadas como lugares de estacionamento.

Uma das opções promissoras a adotar seria as estruturas de estacionamento robótico, totalmente automatizadas. Transportar veículos em elevadores entre os andares é um desafio de engenharia que precisa ser abordado para tornar tais estruturas de estacionamento uma realidade [3]. Um exemplar a salientar é as duas *Car Towers* da *Volkswagen* na *Autostadt* em Wolfsburg, na Alemanha. Cada torre de 48 metros de altura armazena até 400 carros, sendo constituídas cada uma por dois elevadores, onde o carro é transportado numa plataforma com rotação de 180° e armazenado no respetivo depósito [1].

2.1.1 Terminologia dos Elevadores

Existem diferentes tipos de elevadores com diferentes finalidades.

Um **ascensor** é considerado um elevador destinado ao transporte de pessoas ou de pessoas e carga cuja cabina tem dimensões e constituição que permitem o acesso de pessoas.

Um **monta-cargas** é considerado um elevador destinado exclusivamente ao transporte de carga e cuja cabina tem dimensões e constituição que impedem ou dificultam o acesso de pessoas.

De seguida, apresenta-se uma transcrição de partes do Decreto-Lei nº 513/70, de 30 de Outubro, secção Generalidades, sub-secção Definições, Artigo 3º [5], com a descrição dos constituintes de um elevador:

[...]

2. *Ascensor.* - Elevador destinado ao transporte de pessoas ou de pessoas e carga e cuja cabina tem dimensões e constituição que permitem o acesso de pessoas.

5. *Cabina.* - Órgão do elevador onde são transportadas as pessoas ou a carga.

6. *Caixa.* - Local onde se desloca a cabina ou a cabina e o contrapeso.

8. *Casa das máquinas.* - Local destinado à máquina de tração da cabina e aos aparelhos de comando.

9. *Contrapeso.* - Órgão destinado a equilibra o peso da cabina e de parte da sua carga.

10. *Curso.* - Espaço percorrido pela cabina entre os patamares ou níveis extremos.

12. *Elevador.* - Instalação destinada ao transporte de pessoas ou carga entre níveis definidos de serviço numa cabina que se desloca ao longo de guias verticais ou ligeiramente inclinadas sobre a vertical.

13. *Guias.* - Órgãos destinados a guiar o movimento da cabina e do contrapeso.

14. *Limitador de velocidade.* - Dispositivo automático destinado a fazer atuar o pára-quedas no caso de excesso de velocidade.

15. *Máquina de tração.* - Máquina que movimenta a cabina.

19. *Patamar.* - Pavimento ou plataforma onde a cabina estaciona para entrada e saída de pessoas ou carga.

23. *Roda de aderência.* - Roda que por atrito movimenta os cabos de suspensão.

[...]

Os elevadores monta-cargas distinguem-se pelo espaço de instalação e pelo peso da carga a transportar.

As principais características dos elevadores monta-cargas são:

- **Capacidade de transporte:**

- 50 a 100 kg;
- 200 a 500 kg;
- 1500 a 6000 kg;
- 3500 a 5000kg.

- **Curso máximo:**

- 8 metros (hidráulico);
- Entre 12 metros a 22 metros (hidráulico);
- Até 21 metros (até 4 a 6 pisos);
- Até 35 metros (até 12 pisos).

- **Velocidades:**

- Entre 0,1 a 0,6 m/s.

- **Robustez e durabilidade.**

Estes tipos de equipamentos de elevação de cargas podem transportar desde pequenos objetos (monta-pratos para restaurantes ou hospitais) até grandes cargas (elevadores de carga até 5000kg). Para a indústria automóvel, os elevadores do tipo plataforma tesoura ou sistema de estacionamento vertical são também uma boa solução.

Entre os elevadores presentes no mercado, existem diferentes tipos de monta-cargas:

- **Monta-Pratos** – Transporte de pequenas mercadorias de 50 a 100 kg. Utilizados nos setores de restauração, hotelaria, hospitais, farmácias, livrarias;
- **Pequenos Monta-Cargas** – Transporte de mercadorias entre 200 a 500 kg. Aplicação nos setores da indústria, comércio e serviços, como restauração, hotelaria ou lavandarias;
- **Elevadores Monta-Cargas** – Transporte exclusivo de cargas de 200 a 500 kg. Aplicação na indústria de vários setores de atividade;
- **Monta-Cargas Acompanhados** – Transporte em simultâneo de pessoas e cargas, com uma capacidade de transporte que varia entre os 1500 a 6000 kg. Utilizado em meios industriais, sendo ideais em grandes espaços públicos;
- **Monta-Autos Acompanhados** – Equipamentos adequados ao transporte, em simultâneo, de veículos e pessoas, com variação de peso entre 3500 a 5000kg. Utilizados em parques públicos, edifícios residenciais, oficinas ou como substitutos de rampas de acesso.

2.1.2 Legislação no Fabrico de Elevadores

Com a introdução dos ascensores, surgiu também a necessidade de garantir a segurança de pessoas e bens. Vários regulamentos e normas abordam esta temática em várias vertentes, como na conceção dos equipamentos, no fabrico, na instalação, nos ensaios, na manutenção e na utilização destes sistemas.

No Decreto-Lei nº 513/70, de 30 de Outubro [5], estão estabelecidas as condições técnicas a que devem obedecer os elevadores de tração elétrica e cujo objetivo visa essencialmente garantir a segurança e a fiabilidade das instalações.

No Decreto-Lei nº 58/2017, de 9 de junho [6], estão definidos os princípios gerais de segurança a que devem obedecer os ascensores e os respetivos componentes de segurança.

No Decreto-Lei nº 320/2002, de 28 de Dezembro [4], está estabelecido o regime de manutenção e inspeção de ascensores, monta-cargas, escadas mecânicas e tapetes rolantes, após a sua entrada em serviço. Também está estabelecida a transferência da competência de fiscalização e inspeção para as Câmaras Municipais. Os ensaios e exames a efetuar

são realizados de acordo com os aspetos definidos nas normas NP EN 81-1 e 81-2 anexo D.2 [13] para ascensores e na norma EN 81-3 anexo D.2 para monta-cargas.

As normas EN 81-20:2014 e EN 81-50:2014 estão estabelecidas com o objetivo de aumentar as condições de segurança dos utilizadores, dos trabalhadores e das pessoas que possam estar em locais próximos da caixa, ou qualquer local de maquinaria afeta aos ascensores.

2.2 Automação Industrial

2.2.1 Conceito de Simulação

A simulação por computador foi uma ferramenta científica pioneira nas áreas de meteorologia e física nuclear no período imediatamente após a Segunda Guerra Mundial e, desde então, tornou-se indispensável em um número crescente de áreas. A lista de áreas que fazem uso extensivo da simulação por computador cresceu, desde astrofísica, física de partículas, ciência de materiais, engenharia, mecânica dos fluidos, biologia evolutiva, ecologia, economia, teoria da decisão, medicina, sociologia, epidemiologia e muitas outras [33].

A **simulação** é um processo no qual é criado virtualmente uma situação física real. É uma réplica virtual, na qual se consegue estudar o sistema físico. Recorrendo à simulação é possível utilizar um modelo que representa um qualquer sistema e simular o seu comportamento de forma a poder-se efectuar uma avaliação antecipada do seu desempenho.

Atualmente, a economia exige uma reação rápida e flexível ao mercado, com pedidos imprevisíveis de clientes, tornado difícil prever como um sistema ou um processo reagirá, pois torna-se complicado prever isso apenas com cálculos, sendo os sistemas compostos por muitas variáveis dinâmicas [33].

A qualidade, a segurança e a produtividade são todas afetadas pelo processo de simulação. A única ferramenta que pode permitir melhorar e testar esses sistemas complexos e dinâmicos é a **simulação**.

No setor da indústria, projectos na área de controlo e automação dependem essencialmente da simulação do processo e do sistema, que irá permitir determinar as melhores estratégias de controlo e também os melhores parâmetros para estes. Com a simulação é possível testar cenários hipotéticos, evitando assim o risco de ocorrência de avarias nos sistemas reais [34].

Também no ensino académico é importante o uso de simuladores. Recorrendo a simuladores, os alunos podem aprender e testar os seus modelos, sem precisarem de recorrer aos sistemas reais. Com a validação e a verificação dos algoritmos criados são eliminados previamente erros [28], evitando assim que ocorram avarias e problemas nos equipamentos. Os simuladores podem ainda ser utilizados como recurso didático no ensino à

distância.

Um **modelo de simulação** pode ser descrito como um modelo matemático que tem como finalidade a simulação de um sistema real, sendo que este modelo pode ser desenvolvido com a ajuda de um pacote de simulação, para tal é preciso seguir alguns passos [10], nomeadamente:

- Identificação e formulação do problema;
- Obtenção e processamento dos dados do sistema em estudo;
- Formulação e desenvolvimento do modelo;
- Validação do modelo;
- Escrita da documentação do modelo;
- Escolha dos parâmetros de teste;
- Realização da simulação;
- Interpretação dos resultados.

Os **modelos de simulação** podem ser considerados como [10]:

- **Contínuos ou discretos:**
 - Contínuos – O estado do sistema muda continuamente ao longo do tempo, descritos por um conjunto de equações diferenciais;
 - Discretos – O estado do sistema muda em instantes discretos no tempo, em que os eventos estão associados a mudanças de estado.
- **Dinâmicos ou estáticos:**
 - Dinâmicos – O sistema evolui ao longo do tempo;
 - Estáticos – O sistema não se altera ao longo do tempo.
- **Estocástico ou determinístico:**
 - Estocástico – O modelo contém muitos elementos aleatórios, variáveis que têm influenciam no sistema;
 - Determinístico – O modelo não contém elementos aleatórios.

Podem ser utilizados redes de Petri, *Finite State Machines* e linguagens de programação como recurso na criação dos modelos de simulação.

As principais vantagens do uso da simulação são o facto de esta permitir otimizar processos e prever possíveis falhas. A partir da simulação é possível observar os comportamentos do sistema e determinar quais as variáveis mais importantes em ter em conta, bem como estas interagem entre si. A simulação permite ainda estudar o comportamento de um sistema sem o construir, analisando situações improváveis e cenários alternativos.

A simulação, como qualquer outra ferramenta, possui algumas limitações. Os resultados da simulação são por vezes difíceis de interpretar, dificultando a validação dos modelos. O desenvolvimento de um modelo de um sistema, dependendo da sua complexidade, pode ter um custo elevado e levar vários meses para ser elaborado, especialmente nos casos em que os dados são difíceis de obter e pouco coerentes.

2.2.2 Controladores Lógicos Programáveis

O Controlador Lógico Programável (*Programmable Logic Controller* - PLC) revolucionou o setor da automação.

Antes dos PLCs, a única maneira de controlar máquinas era através da lógica cablada, utilizando relés e contactores.

Resumidamente, o relé e o contactor têm um princípio de funcionamento similar. Ambos são compostos por uma bobina que quando energizada, pela tensão de comando, cria um campo electromagnético, que funciona como um íman. Quando o campo é formado, o núcleo move-se, assim como os contactos móveis alteram o seu estado inicial, que são, em geral, normalmente abertos (NA) ou normalmente fechados (NF), fazendo o dispositivo atuar. Quando a corrente da bobina é interrompida, o campo magnético anula-se e os contactos, pela ação da mola, retornam à posição de origem.

Os contactores podem ser classificados como contactores auxiliares e contactores de potência. Os contactores auxiliares e os relés são usados para ligar e desligar circuitos de comando, sinalização e controlo enquanto que, os contactores de potência são usados para o funcionamento dos motores e o dimensionamento destes depende da potência da máquina que vai ser accionada. De referir que os contactores de potência continuam a ser necessários para o arranque e funcionamento das máquinas, o PLC apenas substituiu os circuitos de comando, controlo e auxiliares.

A partir de relés e contactores era possível o controlo de vários motores do mesmo sistema. No entanto, no sector industrial tornou-se complexo, pois as máquinas eram compostas por circuitos eléctricos com muitos encravamentos e, sempre que era necessário fazer alterações ou reparar avarias nos sistemas, esse processo era demorado e dispendioso. Portanto, ficou claro que havia problemas na instalação e manutenção desses grandes sistemas de controlo por relés e contactores [23].

Em resposta aos problemas salientados, em 1968 foi apresentado o primeiro PLC.

Em 1968, a divisão *Hydra-Matic* da *General Motors* emitiu um pedido para a criação de um equipamento eletrônico, capaz de substituir os sistemas de controle por relés. A *General Motors* tinha ainda como requisitos que o equipamento fosse: um sistema flexível como um computador, mas com um preço competitivo com um sistema de lógica de relés; de fácil manutenção e programado com a já aceita linguagem *Ladder*, utilizada na lógica de relés; capaz de funcionar em ambientes industriais, sujeito a poeiras, eletromagnetismo e vibrações; e que fosse modular, permitindo facilidade na troca de componentes e capacidade de expansão [23].

O vencedor da proposta foi a empresa *Bedford and Associates*, resultando desta proposta, o primeiro PLC, chamado de “*Modicon 084*”, por ter sido 84º protótipo desenvolvido na nova empresa Modicon, de Dick Morley, dedicada ao desenvolvimento, fabricação, venda e manutenção deste novo serviço [23].

Dick Morley é então considerado por muitos o “pai” do PLC.

Atualmente, os controladores lógicos programáveis assumem um papel importante em diversas instalações industriais, por serem considerados equipamentos robustos, capazes de operar em ambientes adversos, sujeitos a poeiras e a vibrações, como também pela sua insensibilidade ao ruído elétrico.

Os PLCs são usados na automação de processos eletromecânicos industriais, como, por exemplo, no controle de processos industriais, linhas de montagem, processamento de alimentos ou em armazéns.

Um PLC é composto por:

- Alimentação – Pode ser alimentado por uma fonte de alimentação externa de +24 V DC ou partir da rede elétrica a 230 V AC;
- Unidade Central de Processamento (CPU) – “Cérebro” do PLC, tem a função de ler os valores lógicos nas entradas, executar as instruções que constituem o programa e atualizar na memória o valor das saídas;
- Memória do Autômato – Armazenamento de todas as informações necessárias ao funcionamento do PLC, programa do sistema operativo do PLC, valores das I/O;
- Entradas e Saídas (*Input/Output* - I/O) – Meio de comunicação com dispositivos que fornecem informação sobre o processo e enviam os resultados processados;
- Periféricos – Dispositivos que se conectam ao PLC, para visualização do estado das variáveis através de *software* específico.

A Figura 2.1 ilustra um exemplo de um PLC.



Figura 2.1: Exemplo de um PLC - Modelo Siemens S7-300.

Os PLCs podem ser de vários tamanhos e formas e podem-se classificar como **compactos** (Figura 2.2(a)) ou **modulares** (Figura 2.2(b)).

Os PLCs menores, conhecidos como compactos, são tipicamente projetados com pontos I/O juntos numa mesma unidade.

Os modelos modulares são constituídos por vários módulos de I/O, compostos por módulos de entradas/saídas analógicas e digitais.



(a) PLC compacto [30]



(b) PLC modular [31]

Figura 2.2: Exemplos de PLCs compactos e modulares.

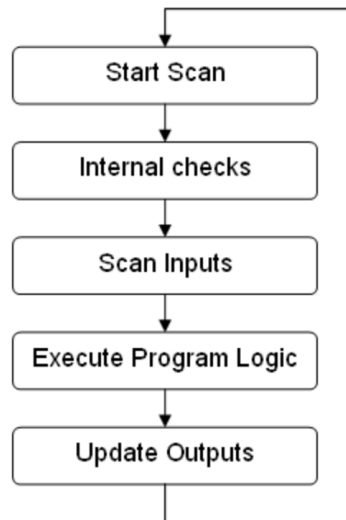


Figura 2.3: *Scan cycle* do PLC [22].

Como se pode observar na Figura 2.3, um *scan cycle* de um PLC consiste em três etapas principais, que envolve a leitura das entradas, a execução da lógica do programa criado pelo utilizador e a atualização das saídas. Este ciclo é repetido várias vezes ao longo da execução do programa do PLC.

Na primeira etapa, a aquisição de entradas implica escrever para a memória do PLC os valores provenientes de sensores. Na etapa de processamento, onde o programa de controlo é executado, é realizado um conjunto de operações aritméticas, com objetivo final de determinar o valor das saídas. Na última etapa, as saídas são atualizadas com os valores obtidos durante o processamento.

No contexto de controlo de um processo industrial, as entradas correspondem, essencialmente, a leituras de valores de sensores e interruptores e as saídas geralmente estão ligadas a actuadores (contactores, relés, actuadores pneumáticos, motores eléctricos e outros equipamentos).

Os valores das entradas, tal como os valores das saídas, podem ser sinais digitais (DI *digital input* e DO *digital output*) ou analógicos (AI *analog input* e AO *analog output*), consoante os módulos instalados no PLC.

2.2.3 Programação de Autómatos

A norma IEC 61131-3 (*International Electrotechnical Commission - Programmable controllers Part 3: Programming languages*) [8, 15] é uma norma internacional onde estão definidas as regras para a escrita das linguagens de programação de autómatos, tendo como propósito promover o desenvolvimento e o uso de *software* compatíveis para diferentes marcas de fabricantes de controladores lógicos programáveis (PLC).

Existem cinco tipos de linguagens de programação de autómatos que são padronizadas pela norma IEC 61131-3:

- **Linguagem *Ladder* (LD)** – Linguagem de programação de PLCs mais comum e a mais difundida. É uma linguagem gráfica que permite descrever uma sequência lógica, baseada na lógica de relés, a partir de símbolos como ligações, temporizadores, contactos e bobines [16];
- **Linguagem *FBD* (*Function Block Diagram*)** – É uma linguagem gráfica baseada na interligação de blocos funcionais, criando um fluxo de dados entre as variáveis de entrada e de saída [17];
- **Linguagem *SFC* (*Sequential Function Chart*)** – Descreve graficamente o comportamento sequencial de um programa de controlo, a partir de passos interligados com blocos de ações e transições. O SFC é a adaptação da norma francesa *Grafcet* [18];
- **Linguagem em Lista de Instruções (IL)** – Consiste num conjunto de instruções (linhas de código) que são executadas de forma sequencial, pelo autómato. As instruções encontram-se agrupadas em sequências [9, 19];
- **Linguagem em Texto Estruturado (ST)** – É uma linguagem para sistemas complexos, sendo uma linguagem estruturada de alto nível, desenhada e projetada para processos de automação. Esta é baseada em escrita de caracteres alfanuméricos, onde uma série de instruções são utilizadas para definir um programa. Permite ainda aceder aos valores das variáveis do processo, como sensores e memórias internas, caso seja necessário, também permite alterar o valor das memórias internas dos PLCs [9, 20].

2.3 Linguagem *Processing*

Processing é uma linguagem de programação de código aberto (*open-source*) e ambiente de desenvolvimento integrado (IDE), concebida e implementada em 2001 por Ben Fry e Casey Reas [25, 26].

2.3.1 História e Características Principais

Ben Fry e Casey Reas eram ambos finalistas no MIT *Media Lab* quando começaram a desenvolver o *Processing*. Na altura eram membros do grupo de computação *Aesthetics and Computation Research*. O *Processing* foi posteriormente desenvolvido nos seus tempos livres enquanto Ben Fry tirava o doutoramento e Casey Reas prosseguia a sua carreira como professor [27].

Ao criar a linguagem *Processing*, Casey Reas e Ben Fry tinham a intenção de construir uma linguagem simples e poderosa para ensinar os fundamentos de programação a artistas plásticos e designers sem as complicações e as enormes barreiras cognitivas que linguagens como C, C++ e Java impunham aos iniciantes [7].

Segundo Casey Reas e Ben Fry, “*Processing é um dialeto de uma linguagem de programação chamada de Java. A sintaxe da linguagem é quase idêntica, mas o Processing adiciona recursos personalizados relacionados com gráficos e interações*” [26].

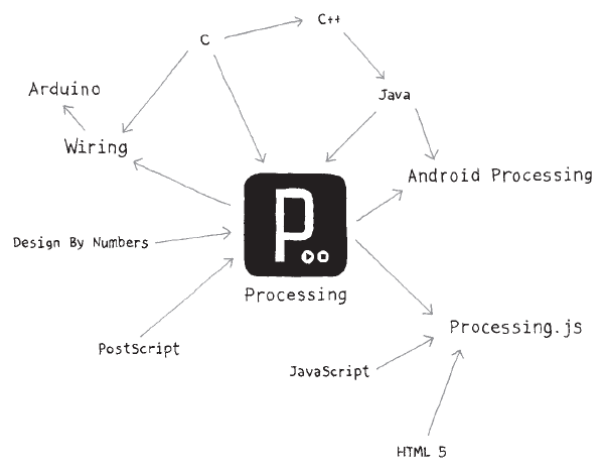


Figura 2.4: O *Processing* está interligado com várias linguagens de programação [26].

Os elementos gráficos do *Processing* estão relacionados com o *PostScript* e o *OpenGL*, ambos utilizados na computação gráfica (Figura 2.4). Assim, compreender o *Processing* pode facilitar a compreensão de outras linguagens de programação, utilizando diferentes ferramentas de *software*, como *Java*, *Android*, *Arduino*, entre outras (Figura 2.4).

A linguagem segue alguns princípios para seu funcionamento. São necessários dois métodos: o *void setup()*, utilizado na execução do *script*; e o *void draw()*, responsável pela repetição do código.

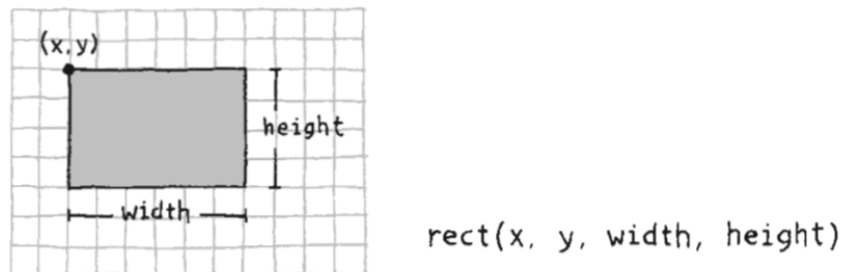


Figura 2.5: Exemplo de como desenhar um retângulo [26].

Para desenhar no *Processing* é preciso definir o tamanho da janela usando a função *size()*, definindo a altura e a largura da janela. A janela funciona como uma tabela pois através das coordenadas dos pontos é possível desenhar (Figura 2.5).

2.3.2 Aplicações

Vários colaboradores e pequenos criadores compartilham os seus programas, contribuem com código e constroem bibliotecas, como o *ControlP5* [29] e o *GP4*. São criadas ferramentas e modos para ampliar as possibilidades do *software*, que são utilizados de forma a facilitar a visualização de dados, o desenho industrial, a criação de imagens para filmes e videoclips, a exportação de imagens para cartazes, revistas e livros, a exportação de arquivos 3D e ainda a programação na área da electrónica.

Visualização de dados e protótipos de *software* são duas das áreas mais desenvolvidas utilizando como ferramenta o *Processing*.

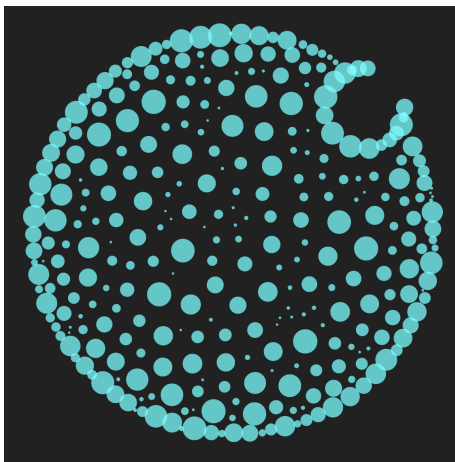
Empresas a nível mundial, como a *Nokia*, a *General Electric* e a *Yahoo!* utilizaram o *Processing* para visualizar os seus dados internos.

Laboratórios de pesquisa de empresas na área da tecnologia, como *Google* e a *Intel*, usaram o *Processing* para criar protótipos de novas interfaces e serviços [27].

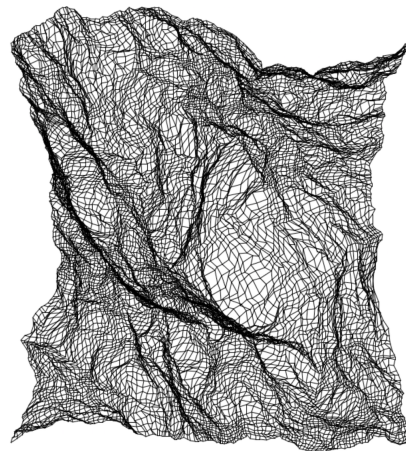
O *Processing* é um *software* utilizado por muitos designers gráficos, artistas e arquitectos. É geralmente utilizado em escolas de arte e em programas de artes visuais nas universidades, mas também é usado com frequência em escolas secundárias e em programas de ciência da computação. Existem diversas maneiras de ensinar a programar, mas os alunos encontram um maior incentivo e motivação quando o *feedback* é visual e imediato [26].

A abordagem do *Processing* foi também aplicada à eletrônica através de projetos em *Arduino* e em *Wiring* (estrutura de programação *open-source* para microcontroladores [2]). O *Arduino* usa uma sintaxe inspirada na mesma que é usada no *Processing* e utiliza uma versão modificada do ambiente de programação *Processing* para ser mais fácil aos alunos aprenderem a programar robôs e também no desenvolvimento de inúmeros outros projetos eletrônicos [27].

Nas Figuras 2.6 e 2.7 são apresentados alguns exemplos de trabalhos desenvolvidos utilizando o *Processing* e também o *Arduino* (Figura 2.7):

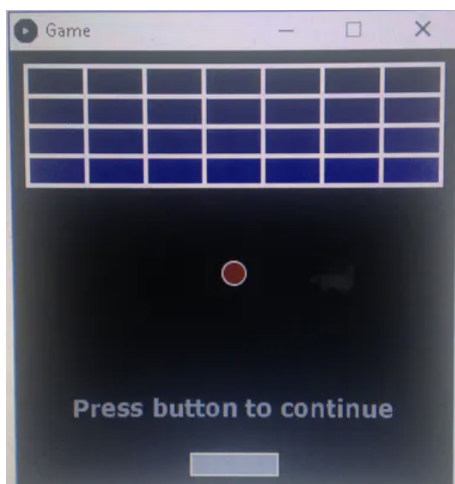


(a) Jelly Slim por nebulaeandstars [11]

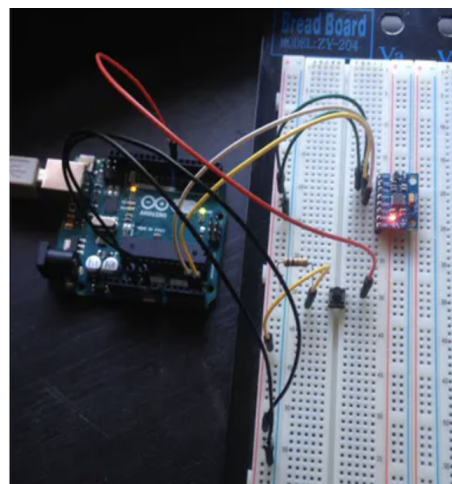


(b) fractalLand por Sabin Serban [32]

Figura 2.6: Exemplos de trabalhos desenvolvidos em *Processing*.



(a) Jogo desenvolvido em *Processing*



(b) O jogo é controlado por um acelerômetro e um botão, que fazem parte do *Arduino*

Figura 2.7: Exemplo de um jogo usando *Processing* e *Arduino* [12].

MODELO PROPOSTO, TECNOLOGIAS E IMPLEMENTAÇÃO

Todo o processo de desenvolvimento do simulador é apresentado e estruturado neste capítulo.

O presente capítulo está dividido em três partes.

Na primeira parte são apresentados o conjunto de procedimentos e métodos a utilizar na realização deste trabalho.

Na segunda parte são descritas todas as funcionalidades do simulador juntamente com todas as características aplicadas a este, desde os modos de funcionamento do elevador até ao tipo de animação aplicada ao simulador.

Por último é apresentada a interface gráfica do simulador.

3.1 Desenvolvimento do Simulador

3.1.1 Metodologia Proposta

Para o desenvolvimento do simulador de um elevador do tipo monta-cargas em linguagem *Processing* foi necessário definir um conjunto de procedimentos e métodos a utilizar.

De seguida são descritas quatro das principais etapas tidas em conta na realização deste trabalho.

1. Compreensão do Funcionamento de um Elevador Monta-Cargas

Para o desenvolvimento do simulador é necessário analisar as características específicas dos elevadores e definir quais destas a adotar. Como este simulador é baseado no protótipo de elevador monta-cargas que se encontra no laboratório 2.3 do departamento de Engenharia Electrotécnica e de Computadores, é igualmente necessário analisar os constituintes deste protótipo para se estabelecer quais as características importantes a aplicar ao simulador.

2. Desenvolvimento do Simulador

Esta segunda etapa é provavelmente a etapa mais complexa e a mais importante deste trabalho. A interface gráfica para o utilizador (GUI) do simulador é desenvolvida em linguagem *Processing*. Para se proceder ao desenvolvimento do simulador é necessário perceber como é o ambiente e a sintaxe da linguagem *Processing*.

Inicialmente é criada uma interface simples para testar o simulador em ambiente *Processing*, com apenas 3 botões e 3 sensores de proximidade com a função de enviar e detetar a cabina no piso 0, piso 1 e piso 2. Na versão final do simulador são implementadas outras funcionalidades, como sequências de funcionamento, sensores e fins de curso.

3. Aplicação da Linguagem em Lista de Instruções (IL)

Nesta etapa é feito um estudo para melhor entender a linguagem IL e como pode ser associada ao modo de funcionamento do simulador.

4. Controlo do Simulador

Neste ponto são definidos os modos de funcionamento do elevador. O elevador monta-cargas pode ser controlado em modo manual ou em modo automático.

Em modo manual, o elevador é controlado por um conjunto de 3 botões. Num sistema real, este conjunto estaria colocado no interior da cabina do elevador.

No modo automático, o elevador pode ser controlado por um conjunto de 3 sequências de funcionamento, em que num sistema real, este conjunto estaria situado no exterior junto ao elevador, no piso 0.

3.1.2 Tecnologias Utilizadas

Sendo o simulador baseado no protótipo de um elevador monta-cargas, foi necessário analisar e compreender o funcionamento desse elevador.

3.1.2.1 Descrição do Protótipo de Elevador do tipo Monta-Cargas

Na Figura 3.1 está representado o protótipo de elevador do tipo monta-cargas, desenvolvido no âmbito da tese de mestrado do aluno Nuno Ângelo, intitulada "*Projecto e Concepção de um Sistema Elevador Monta-Cargas Industrial*" [35]. Em seguida, são apresentadas as especificações técnicas do protótipo.

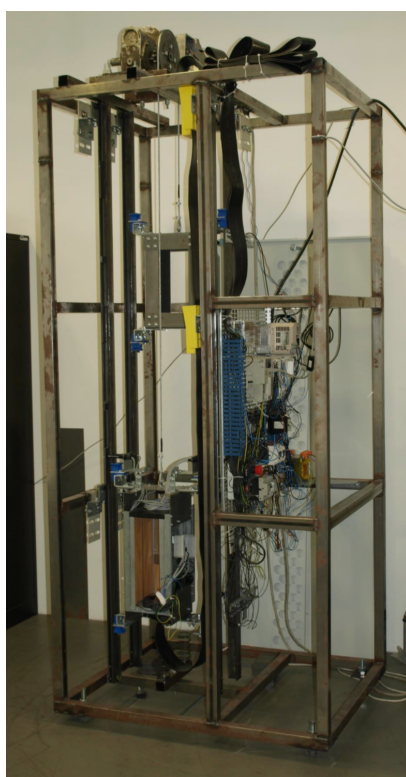


Figura 3.1: Protótipo de elevador do tipo monta-cargas.

O protótipo de elevador do tipo monta-cargas (Figura 3.1) está montado numa estrutura com $(1.08 \times 0.88 \times 2.30)$ metros e serve três patamares definidos por P0, P1 e P2. Como se pode observar na Figura 3.1, a estrutura do elevador é constituída pela cabina, o contrapeso, dois sensores mecânicos de fim de curso (superior e inferior), quatro sensores magnéticos de proximidade, duas guias, amortecedores e pela máquina de tração.

É ainda composto, como representado na Figura 3.2, pelo PLC, o variador de velocidade, a botoneira e alguns atuadores e proteções elétricas.

A máquina de tração é um motor trifásico cuja função é imprimir rotação a uma roda de aderência, sendo o sentido de rotação definido pelo PLC e imposta pelo variador de velocidade.

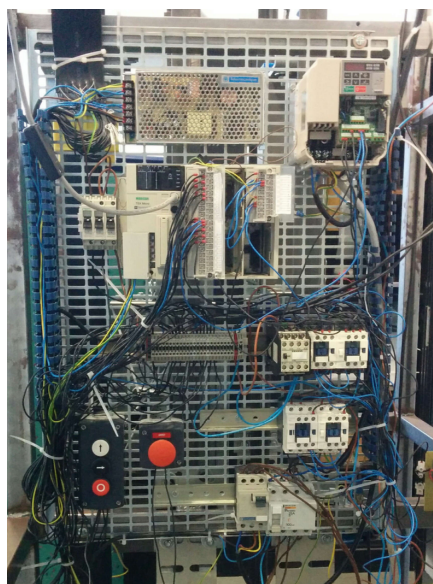


Figura 3.2: Equipamentos da parte de controlo do elevador.

Estão definidas um conjunto de especificações do protótipo de elevador, que são apresentadas de seguida.

3.1.2.2 Especificações Funcionais

Estão definidos três níveis de patamar - P0, P1 e P2 - onde, P0 significa piso 0, P1 significa piso 1 e P2 significa piso 2. A supervisão e o controlo do automatismo é realizado pelo controlador lógico programável (PLC) que comanda o variador de frequência (VF) e este, por sua vez, o motor elétrico.

Para se ligar o elevador inicia-se o ciclo do seguinte modo:

- Existem quatro botões, B0, B1, B2 e PE, onde:
 - **Botão B0** - Permite enviar a cabina para o patamar P0, caso não se encontre inicialmente nesse patamar;
 - **Botão B1** - Permite enviar a cabina para o patamar P1;
 - **Botão B2** - Permite enviar a cabina para o patamar P2;
 - **Botão PE** - Permite imobilizar imediatamente o elevador.
- Estando a cabina no patamar P0, é possível escolher entre duas sequências de funcionamento:
 - **Sequência S1** - Premindo o botão B1;
 - **Sequência S2** - Premindo o botão B1 e de seguida o botão B2.
- Após a seleção da sequência, a cabina só arranca ao fim de quatro segundos;

- Após a chegada da cabina ao patamar P1 ou P2, a cabina permanece no patamar durante quatro segundos até arrancar para o próximo patamar;
- A paragem imediata da cabina pode também resultar da atuação dos fins de curso (fim de curso inferior (FCI) e fim de curso superior (FCS)) quando a cabina ultrapassa os patamares P0 ou P2, resultando na paragem da máquina de tração e o bloqueio do programa de controlo PLC.

Na estrutura da cabina existem quatro sensores magnéticos (AF, AT, BF e BT), que permitem a deteção da posição exata da cabina ao longo do curso.

3.1.2.3 Especificações Tecnológicas

Dentro da tecnologia utilizada, foram usadas as seguintes:

- Máquina de Tração – Motor elétrico de acionamento trifásico. A máquina de tração faz mover a cabina, quando a bobina do travão (freio) eletromecânico está energizada, libertando a máquina de tração, permitindo, assim, o movimento da cabina. Quando a bobina deixa de estar energizada, o travão trava o rotor do motor elétrico, impedindo-o de se movimentar;
- Controlador Lógico Programável (PLC) – Modelo TSX Micro 3721 e o *software* de programação é o PL7 MICRO. O PLC envia sinais analógicos em tensão contínua para o variador de frequência, permitindo alterar a velocidade do motor elétrico;
- Foram utilizados três tipos de linguagem de programação: linguagem em texto estruturado (ST), linguagem SFC juntamente com a linguagem em lista de instruções (IL) (SFC/IL);
- Variador de frequência – Modelo OMRON V1000, constituído por uma consola de interface homem-máquina (HMI);
- Os quatro sensores (AF, AT, BF e BT) funcionam como sensores de posicionamento magnético, sendo que os sensores AF e AT são contactos do tipo NA e os sensores BF e BT são biestáveis;
- Os botões B0, B1, B2 e PE são botões de pressão (contactos elétricos instantâneos), sendo que os botões B0, B1 e B2 são do tipo NA e o botão PE é do tipo NF.

3.1.2.4 Especificações Operacionais

O variador de frequência (VF) é composto por uma consola de interface homem-máquina (HMI) que permite visualizar e parametrizar os parâmetros do variador, sendo possível ao operador aceder a alguns dos dados em tempo real e controlar o motor elétrico da máquina de tração localmente.

Os fins de curso superior (FCS) e inferior (FCI) atuam antes da cabina ou do contrapeso atingirem os respetivos amortecedores, sendo acionados mecanicamente pelo contrapeso.

O botão B0 permite reinicializar o automatismo após a atuação de uma segurança ou falha de energia elétrica. Em caso de engano na escolha da sequência de funcionamento, é necessário premir o botão B0 e voltar a selecionar a sequência pretendida.

3.1.2.5 Sensores de Posição e respetivos Ímanes

Os sensores do protótipo de elevador são interruptores magnéticos de proximidade do tipo *reed switch* (Figura 3.3) que, quando passam em frente a um íman, os contactos respetivos abrem ou fecham.

Estão localizados no topo da estrutura da cabina e funcionam com tensão de + 24 V DC.

No anexo 3, a Figura I.4 apresenta a disposição dos sensores e dos ímanes ao longo das guias do protótipo de elevador do tipo monta-cargas.



(a) Sensor AF e AT, do lado esquerdo da cabina



(b) Sensor BF e BT, do lado direito da cabina

Figura 3.3: Sensores do protótipo.

Características dos sensores AF e AT:

- Estão montados do lado esquerdo da cabina (tendo como referência o operador em frente a esta - Figura 3.3(a));
- O sensor AF está colocado à frente e o sensor AT está colocado atrás (Figura 3.3(a));
- São sensores monoestáveis;
- Cada sensor possui um contacto NA, que fecha quando passa em frente ao respetivo íman, sendo o sinal da posição da cabina enviado ao PLC;
- Quando os sensores AF e AT são atuados simultaneamente pelo efeito do campo magnético dos ímanes, significa que a cabina se encontra ao nível do patamar P0, P1 ou P2, ou parada ou imediatamente a sair ou a chegar ao patamar;
- Nos movimentos de subida e descida da cabina, os sensores AF e AT são também responsáveis pela mudança de velocidade da cabina;
 - Os sensores AT (no movimento de subida) e AF (no movimento de descida) servem para:
 - * Mudar da velocidade lenta para a velocidade rápida, quando a cabina sai do patamar em que se encontrava;
 - * Mudar da velocidade rápida para a velocidade lenta antes de chegar ao patamar destino.

Características dos sensores BF e BT:

- Estão montados do lado direito da cabina (tendo como referência o operador em frente a esta - Figura 3.3(b));
- O sensor BF está colocado à frente e o sensor BT está colocado atrás (Figura 3.3(b));
- São sensores biestáveis, com efeito de "memória", que ao passarem pelo respetivo íman, os contactos são atuados e mantêm o estado mesmo depois de já terem passado o íman, no mesmo sentido de movimento;
- A cada sensor corresponde um íman bipolar;
- Os sensores BF e BT informam o PLC se a cabina se encontra no patamar P0 ou abaixo deste, entre os patamares P0 e P2, no patamar P2 ou acima deste;
- Quando ambos os contactos dos sensores BF e BT estão fechados, significa que passa corrente elétrica em cada um deles e a cabina está entre os patamares P0 e P2;

- Com o contacto do sensor BF aberto, não passa corrente através deste, sendo que o PLC recebe a informação que a cabina:
 - Possui movimento de subida;
 - Está no patamar P0 ou abaixo desse.
- Com o contacto do sensor BT aberto, não passa corrente através deste, sendo que o PLC recebe a informação que a cabina:
 - Possui movimento de descida;
 - Está no patamar P2 ou acima desse.

3.1.2.6 Fim de Curso Inferior (FCI) e Fim de Curso Superior (FCS)

Os fins de curso de segurança têm como função a paragem da máquina de tração no caso da cabina exceder o seu curso normal de funcionamento (entre P0 e P2).

Estão montados na guia direita e são acionados mecanicamente pelo contrapeso.

Os fins de curso têm incorporado um contacto elétrico NF, que promove a paragem da cabina quando actuado.

Após a actuação do fim de curso inferior ou superior, só é possível a reposição em serviço do elevador quando premido o botão B0 (*reset*).

No anexo 3, na Figura I.4 está representada a localização dos fins de curso do protótipo de elevador do tipo monta-cargas.

3.1.2.7 Tabela de Endereços do PLC TSX Micro 3721

O modelo do PLC utilizado é o TSX Micro 3721 e o *software* de programação é o PL7 MICRO.

O PLC foi programado utilizando a linguagem de texto estruturado (ST) e os diagramas implementados foram elaborados em SFC/IL.

Os diagramas elaborados em SFC foram utilizados para envio de sinais digitais para os atuadores (KM0, KM1, KF2 e KF3).

O contactor de descida (KM0) é alimentado a uma tensão de + 24 V DC, quando o contacto NA do relé KM0 fecha, pertencente ao módulo TSX DMZ 28 DR. O fecho de KM0 dá ordem de movimento de descida à cabina.

O contactor de subida (KM1) é alimentado a uma tensão de + 24 V DC, quando o contacto NA do relé KM1 fecha, pertencente ao módulo TSX DMZ 28 DR. O fecho de KM1 dá ordem de movimento de subida à cabina.

O código implementado em texto estruturado (ST) permite “ler” a informação proveniente de botões e sensores, enviar sinais analógicos do PLC para o variador de frequência do motor elétrico e receber sinais analógicos de monitorização deste para o PLC.

O PLC TSX Micro 3721 é composto pelo módulo TSX DMZ 28 DR (módulo de entradas/saídas digitais) e pelo módulo TSX AMZ 600 (módulo de entradas/saídas analógicas). O tempo de ciclo “*scan cycle*” do PLC TSX Micro 3721 é de 5 milissegundos, isto é, executa 200 ciclos em 1 segundo.

O sinal das entradas e saídas digitais (DI e DO) são a + 24 V DC.

O sinal das entradas e saídas analógicas (AI e AO) são de 0 a + 10 V DC.

No anexo 1, encontra-se a tabela I.1, na qual são apresentadas todas as entradas e saídas disponíveis no PLC. No anexo 2, encontra-se os pontos característicos do PLC TSX Micro 3721 (Figura I.1) e a localização do módulo DMZ 28 DR (Figura I.2).

3.2 Especificação do Simulador

3.2.1 Descrição do Funcionamento do Simulador

Como foi referido anteriormente, este trabalho tem como objetivo o desenvolvimento de um simulador de um elevador do tipo monta-cargas, em linguagem *Processing*.

Este simulador será capaz de, através de uma interface acessível, permitir ao utilizador verificar e testar as diferentes ações possíveis para a um modelo de elevador do tipo monta-cargas.

A partir do simulador é possível estudar a linguagem lista de instruções (IL), aplicada ao movimento do elevador.

Na Figura 3.4 está representada a interface gráfica do simulador em linguagem *Processing*.

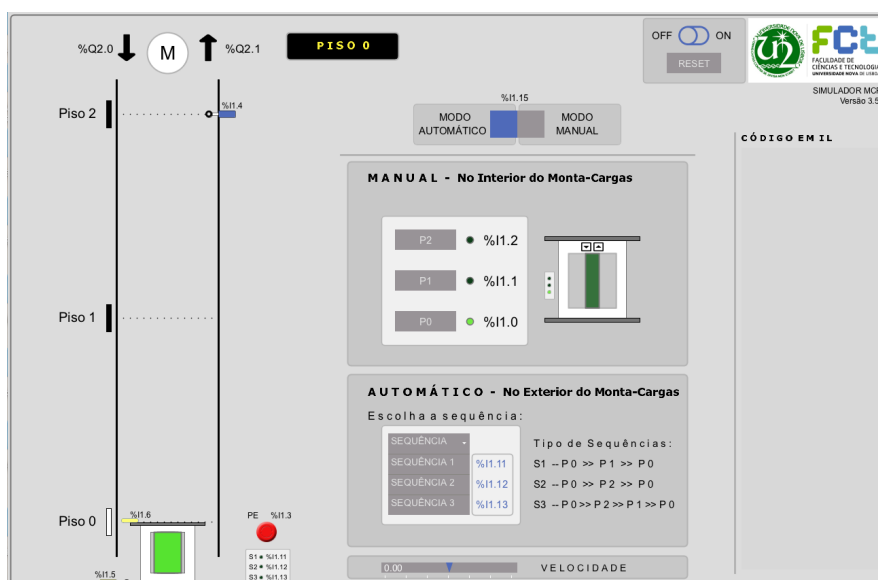


Figura 3.4: Interface do simulador de elevador do tipo monta-cargas.

Em seguida é apresentada uma breve descrição do funcionamento do simulador.

Na Figura 3.5 está representado, de forma simplificada, o funcionamento do simulador de elevador do tipo monta-cargas.

Na estrutura do elevador estão definidos 3 pisos - Piso 0, Piso 1 e Piso 2. O elevador, inicialmente, encontra-se no piso 0.

O elevador pode ser controlado através de dois modos: modo manual e modo automático (Figura 3.5).

Se o utilizador do simulador selecionar o modo de funcionamento manual, pode escolher entre três botões: botão P0, envia o elevador para o piso 0; botão P1, envia o elevador para o piso 1; e botão P2, envia o elevador para o piso 2.

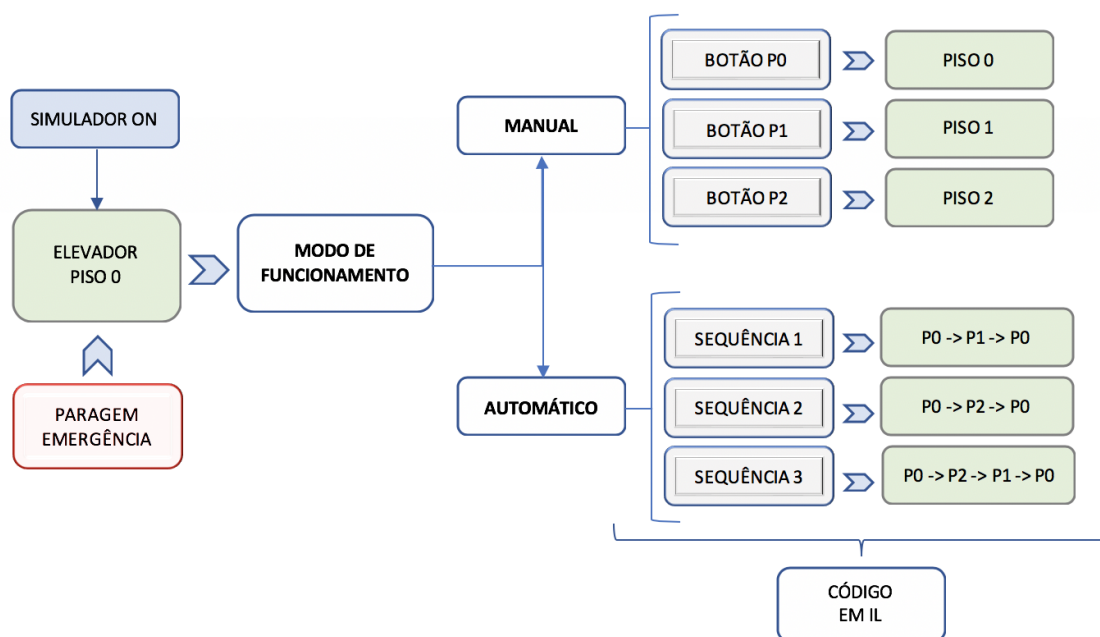


Figura 3.5: Funcionamento do simulador.

Se o utilizador seleccionar o modo automático, pode escolher entre três sequências de funcionamento: sequência 1 – P0 » P1 » P0; sequência 2 – P0 » P2 » P0; e sequência 3 – P0 » P2 » P1 » P0.

Quando é escolhida uma sequência no modo automático, é possível visualizar a sequência seleccionada na forma de código em linguagem lista de instruções (IL) e ir acompanhando o ciclo de funcionamento dessa sequência.

Caso o botão de paragem de emergência (PE) seja pressionado, o elevador é imobilizado e é enviado para o piso 0. Também é possível visualizar o código IL do botão PE.

3.2.2 Autómata Finito aplicado ao Funcionamento do Simulador

Autómata Finito (*Finite Automata* (FA)) ou máquina de estados finitos (*Finite States Machines*), representada através de diagrama de estados, define-se como um conjunto finito de estados, sendo um deles definido como estado inicial, composto por transições, que correspondem à mudança de um estado de origem para um estado destino (que pode ser o mesmo que estado de origem) e ainda por um conjunto de entradas e regras específicas para a transição entre estados [21].

Do ponto de vista Autómata Finito, podemos pensar no simulador de elevador montado com cargas como uma máquina de estados. Quando se carrega num botão, o elevador vai-se mover de um estado para outro, por exemplo, do piso 0 para o piso 1.

A transição acontece quando é pressionado um botão.

Na Figura 3.6 é apresentado um exemplo de Autômata Finito associado aos modos de funcionamento do simulador.

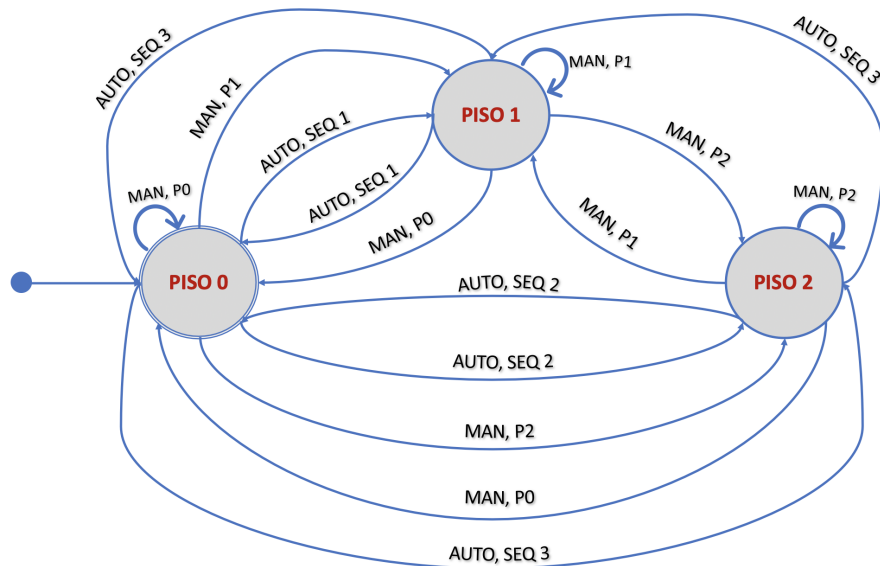


Figura 3.6: Exemplo de Autômata Finito do modelo do simulador.

Como se pode constatar pela Figura 3.6, existem três estados - Piso 0, Piso 1 e Piso 2.

Em relação às transições, estas estão associadas aos dois modos de funcionamento do simulador: modo manual (MAN) e modo automático (AUTO).

As transições do modo manual correspondem à escolha do botão P0, P1, ou P2, sendo que:

- **MAN, P0** – Botão P0 – envia a cabina para piso 0;
- **MAN, P1** – Botão P1 – envia a cabina para piso 1;
- **MAN, P2** – Botão P2 – envia a cabina para piso 2.

As transições do modo automático correspondem à seleção da sequência pretendida, em que:

- **AUTO, SEQ 1** – Sequência 1 – Piso 0 » Piso 1 » Piso 0;
- **AUTO, SEQ 2** – Sequência 2 – Piso 0 » Piso 2 » Piso 0;
- **AUTO, SEQ 3** – Sequência 3 – Piso 0 » Piso 2 » Piso 1 » Piso 0.

Portanto, a transição “AUTO, SEQ 1” corresponde à seleção da sequência 1, que vai gerar um evento, no qual o elevador se desloca do piso 0 para o piso 1 e, após 4 segundos, do piso 1 para o piso 0.

Nas Figuras 3.7 e 3.8 estão representadas as máquinas de estados finitos para o modo manual e para o modo automático.

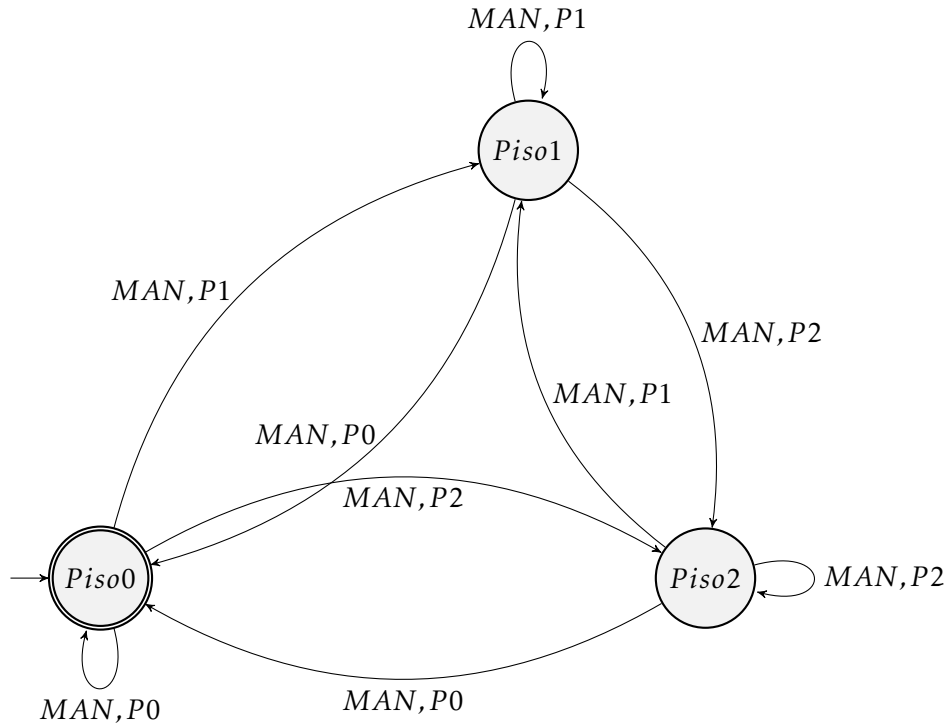


Figura 3.7: Máquina de estados finitos do modo manual.

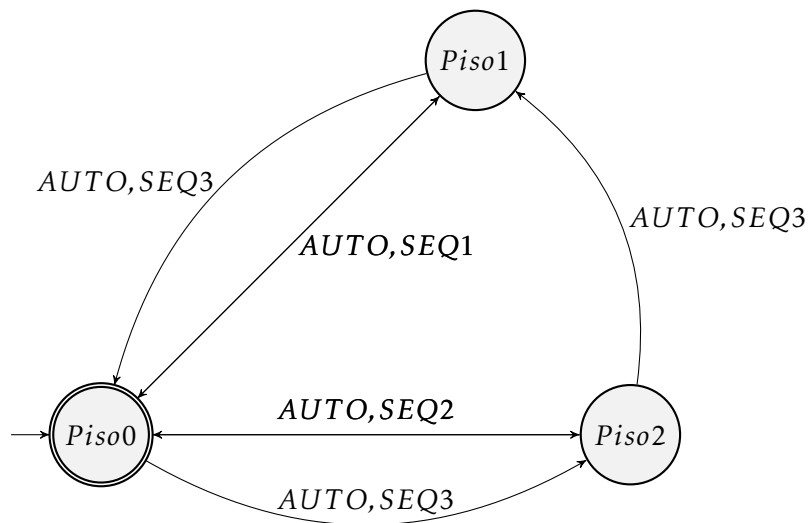


Figura 3.8: Máquina de estados finitos do modo automático.

3.2.3 Especificações funcionais

Para o desenvolvimento do simulador foi necessário analisar as características dos elevadores e também perceber quais das características do protótipo de elevador do tipo monta-cargas poderiam ser aplicadas ao simulador.

No decorrer do desenvolvimento deste trabalho estabeleceram-se quais as principais funcionalidades e características necessárias a implementar.

O **modelo gráfico do elevador** foi criado com base em algumas das características do protótipo de elevador monta-cargas. Foram consideradas apenas as características mais relevantes para este trabalho, tendo sido adaptadas ao modelo do simulador.

No modelo gráfico do elevador estão incluídos alguns componentes, como:

- Cabina do elevador;
- Guias e estrutura;
- Sensor e ímanes;
- Fins de curso;
- Símbolo da máquina de tração;
- Identificação dos pisos e do estado do sistema.

Para o **controle do simulador** foram estabelecidos dois modos de funcionamento, e para tal foram utilizados:

- Botão selector modo automático/manual;
- Para o modo manual:
 - **Botão P0** – envia a cabina para piso 0;
 - **Botão P1** – envia a cabina para piso 1;
 - **Botão P2** – envia a cabina para piso 2.
- Para o modo automático:
 - **Sequência 1** – Piso 0 » Piso 1 » Piso 0;
 - **Sequência 2** – Piso 0 » Piso 2 » Piso 0;
 - **Sequência 3** – Piso 0 » Piso 2 » Piso 1 » Piso 0.

Para o caso de ocorrerem falhas de segurança ou avarias foi incluído um botão de paragem de emergência (botão PE).

Na interface gráfica do simulador existe ainda um botão que possibilita variar a velocidade do elevador.

De forma a visualizar os **estados e comportamentos** do simulador, recorreu-se a:

- **Sensor e ímanes** – Detetam a posição do elevador;
- **Fins de curso** – Detetam se o elevador se encontra no piso 0 ou no piso 2 e também no caso de exceder o seu curso;
- **Leds** – Identificam os pisos selecionados e a posição do elevador;
- **Imagens com animação** - Identificam o movimento e estado do elevador;
- **Ecrã de informações** - Visualizar o estado do elevador no momento.

O simulador pode ser utilizado como recurso didático no ensino de linguagens de programação, como a linguagem lista de instruções (IL).

Para cada sequência de funcionamento está associado um **código em linguagem lista de instruções (IL)**.

De forma a que o utilizador consiga acompanhar o funcionamento do elevador a partir do código em IL, foi acrescentado à interface gráfica:

- **Endereços das entradas e saídas do sistema** – Permite saber qual o endereço de cada componente;
- **Código em IL** – Espaço para o código em lista de instruções.

3.2.4 Arquitetura do Sistema

A arquitetura com a metodologia proposta encontra-se representada na Figura 3.9.

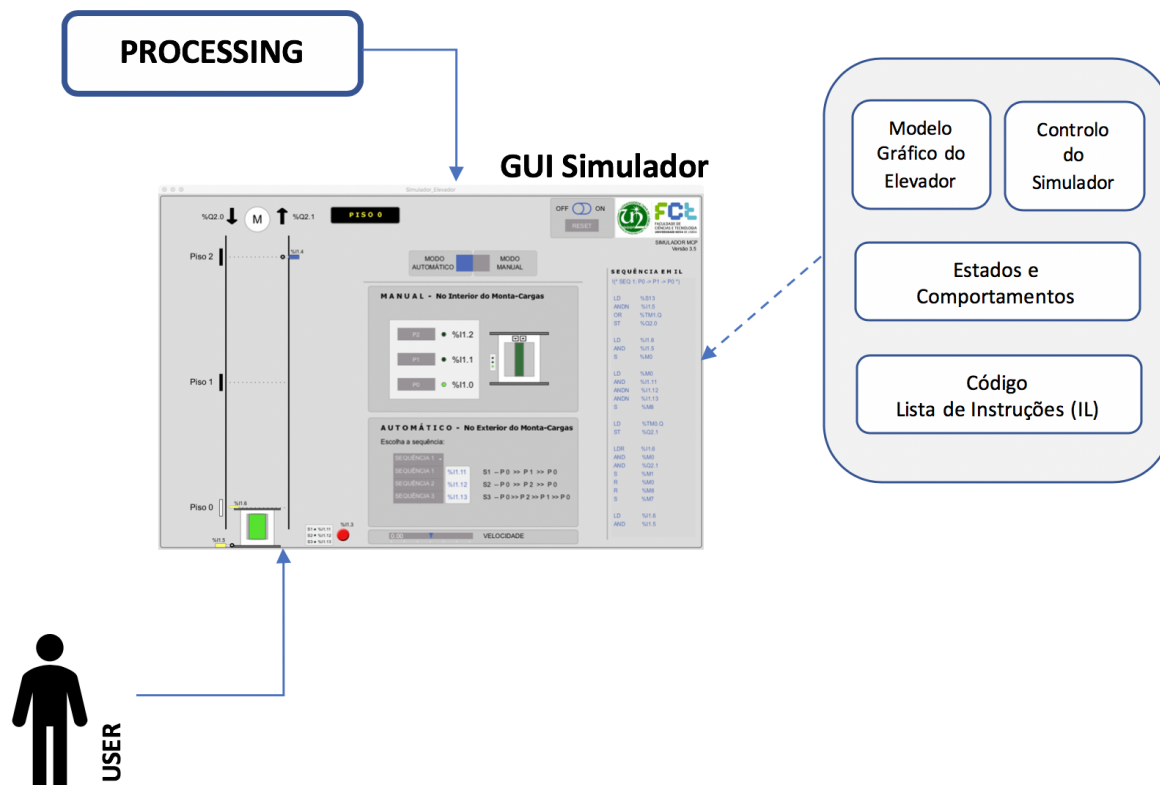


Figura 3.9: Arquitetura da metodologia proposta.

A interface gráfica do simulador é composta pelo modo de controlo do simulador, permitindo ao utilizador controlar o elevador com recurso a botões e selectores, sendo possível visualizar o estado do sistema e ainda acompanhar todo o processo de funcionamento do elevador.

3.3 Implementação do Simulador

3.3.1 Mecanismos de Simulação

O simulador de elevador do tipo monta-cargas foi desenvolvido em linguagem *Processing*.

Este simulador é composto por botões e sensores que permitem ativar as várias funcionalidades do elevador.

É possível observar a estrutura da interface gráfica do simulador na Figura 3.10.

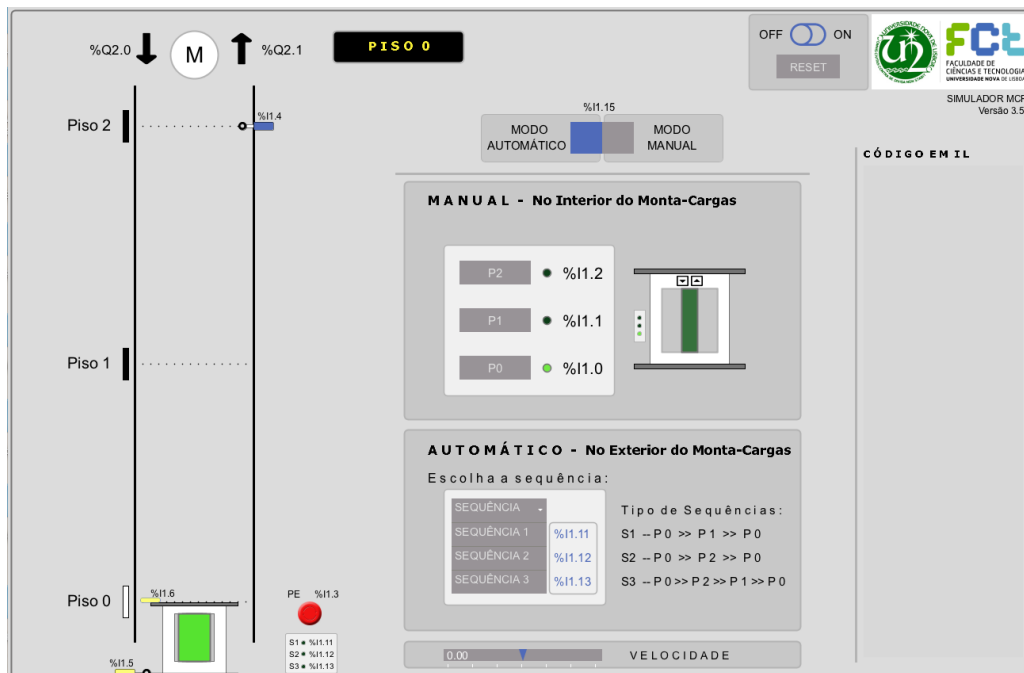


Figura 3.10: Interface gráfica do simulador.

3.3.1.1 Botões

De forma a conceder funcionalidade ao simulador foram utilizados botões, desenvolvidos a partir de bibliotecas e imagens.

Para alguns botões foi utilizada a biblioteca *ControlP5*. O *ControlP5* foi desenvolvido por *Andreas Schlegel* e oferece uma variedade de controladores que permitem facilmente alterar e ajustar variáveis enquanto o *sketch* está em execução [29].

Botão OFF/ON e RESET

O botão OFF/ON é usado para “ligar” o simulador. Se não for ligado, o elevador não se movimenta.

O botão *RESET* é utilizado para reiniciar o simulador, repondo o elevador no estado inicial (piso 0).

Botão selector Modo Automático/Manual

Foram definidos 2 modos de funcionamento para o elevador:

- Modo Manual;
- Modo Automático.

O botão modo automático/manual permite seleccionar o modo de funcionamento do elevador em estudo.

Inicialmente, o modo automático está activo, como se o elevador estivesse a funcionar em modo remoto.

Controlo do Modo Manual

As principais características do modo manual são as seguintes:

- Conjunto de 3 botões - P0, P1 e P2:
 - **Botão P0** – envia a cabina para piso 0;
 - **Botão P1** – envia a cabina para piso 1;
 - **Botão P2** – envia a cabina para piso 2.
- Permite seleccionar o piso desejado;
- A cabina só vai para o piso seleccionado após 4 segundos;
- Num sistema real este conjunto de botões estaria localizado no interior da cabina;
- Em caso de avaria ou necessidade de manutenção, pode ser utilizado como recurso para manobrar o elevador.

Controlo do Modo Automático

As principais características do modo automático são as que se apresentam em seguida:

- Módulo de seleção da sequência - Sequência 1, Sequência 2 e Sequência 3:
 - **Sequência 1** – envia a cabina do Piso 0 » Piso 1 » Piso 0;
 - **Sequência 2** – envia a cabina do Piso 0 » Piso 2 » Piso 0;
 - **Sequência 3** – envia a cabina do Piso 0 » Piso 2 » Piso 1 » Piso 0.
- Permite seleccionar a sequência pretendida;

- Para cada sequência de funcionamento está associado um código em IL;
- Num sistema real, este conjunto estaria localizado no exterior da cabina, junto ao elevador.

Botão de Paragem de Emergência

Caso o botão de paragem de emergência (PE) seja pressionado, a cabina pára e, após 4 segundos, é enviada para o piso 0 (piso inicial).

Controlo da Velocidade

Na interface do simulador existe ainda um botão do tipo “*slider*” que permite alterar a velocidade de deslocamento do elevador.

Na Figura 3.11 é apresentado um resumo dos botões utilizados na interface gráfica do simulador.



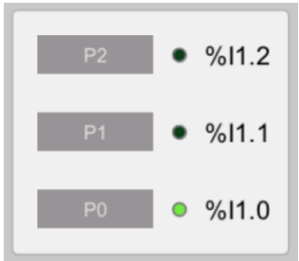
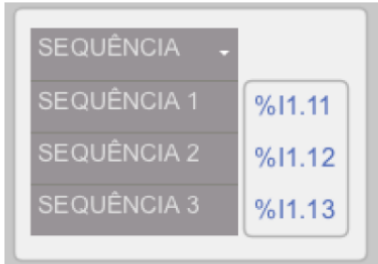
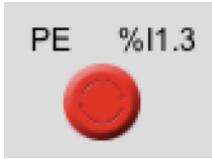
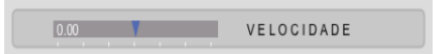
BLOCOS	BOTÕES	FUNÇÃO
	OFF/ON RESET	<ul style="list-style-type: none"> Ligar simulador Reiniciar simulador
	MODO AUTO/MANUAL	<ul style="list-style-type: none"> Selecionar modo de funcionamento Modo automático ativo
	MODO MANUAL PISO 2 PISO 1 PISO 0	<ul style="list-style-type: none"> Botão P0 – envia o elevador para o piso 0 Botão P1 – envia o elevador para o piso 1 Botão P2 – envia o elevador para o piso 2
	MODO AUTOMÁTICO SEQUÊNCIA 1 SEQUÊNCIA 2 SEQUÊNCIA 3	<ul style="list-style-type: none"> Seq. 1 – P0 >> P1 >> P0 Seq. 2 – P0 >> P2 >> P0 Seq. 3 – P0 >> P2 >> P1 >> P0
	PARAGEM DE EMERGÊNCIA	<ul style="list-style-type: none"> Pára o elevador quando é acionado
	VELOCIDADE	<ul style="list-style-type: none"> Botão “slider” controla a velocidade do elevador

Figura 3.11: Resumo dos botões utilizados no simulador.

3.3.1.2 Sensor e Fins de Curso

Ao modelo gráfico do elevador foram adicionadas algumas características interessantes, de forma a conceber animação ao simulador.

Como referido anteriormente, algumas das características e funcionalidades do protótipo de elevador do tipo monta-cargas foram utilizadas no simulador.

Sensor AF

O sensor AF, como referido na secção 3.1.2.5, funciona como um interruptor de tipo magnético de proximidade que, quando passa em frente ao íman, o contacto fecha. Deste modo é possível detetar a posição actual da cabina ao longo do curso.

Fim de Curso Inferior e Superior

O fim de curso inferior (FCI) tem como função detetar se a cabina se encontra no piso 0, ou abaixo deste, no caso da cabina exceder o seu curso normal.

No piso 2, encontra-se o fim de curso superior (FCS) que permite detetar se a cabina se encontra no piso 2, ou acima deste.

Como referido na secção 3.1.2.6, os fins de curso têm incorporado um contacto elétrico NF que, quando actuado, promove a paragem da cabina.

O FCI e o FCS, que fazem parte do modelo gráfico do elevador, são do tipo interruptores de fim de curso acionados por uma alavanca de rolo.

3.3.2 Visualização e Animação

De forma a tornar o simulador mais interativo e apelativo para o utilizador, foram incluídas algumas características e respostas às rotinas dos botões.

Em seguida vão ser ilustradas algumas das respostas visuais.

Na Figura 3.12 é apresentada a resposta visual do sensor AF e dos fins de curso (FCI e FCS).

As respostas visuais ao botão de paragem de emergência (PE) e ao botão OFF/ON estão ilustradas na Figura 3.13.

Na Figura 3.14 é possível visualizar algumas das característica adicionadas ao simulador de forma a que o estado do elevador, num determinado momento, seja perceptível.

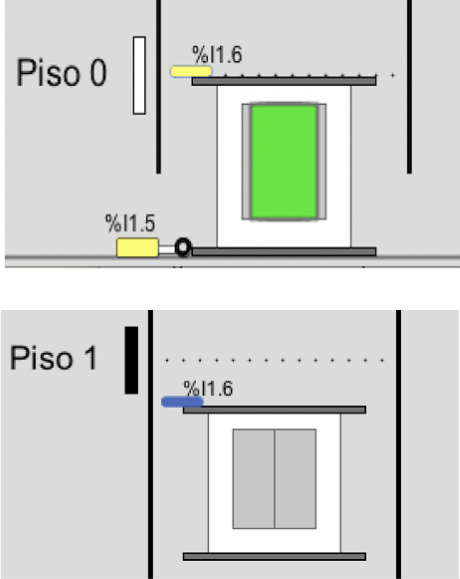
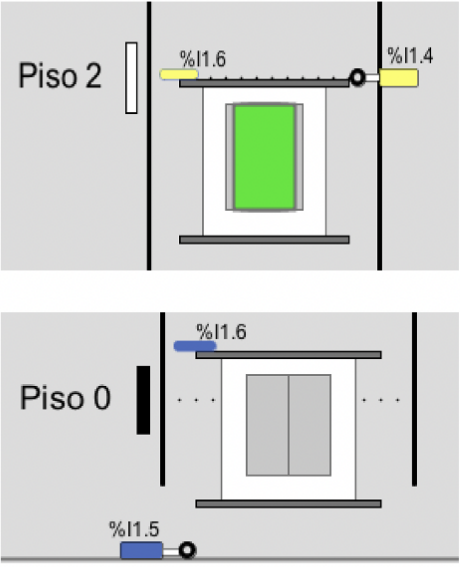
VISUALIZAÇÃO	ESTADO
	<p data-bbox="1007 477 1182 506">SENSOR AF</p> <p data-bbox="1054 533 1134 562">%I1.6</p> <ul data-bbox="842 651 1406 887" style="list-style-type: none"><li data-bbox="842 651 1406 752">▪ Quando o sensor AF passa pelo íman, o sensor fica amarelo e o íman a branco;<li data-bbox="842 786 1406 887">▪ Quando o sensor AF deixa de detetar o íman, o sensor fica azul e o íman a preto.
	<p data-bbox="999 1189 1190 1218">FCI e FCS</p> <p data-bbox="999 1245 1190 1274">%I1.5 e %I1.4</p> <ul data-bbox="829 1364 1406 1532" style="list-style-type: none"><li data-bbox="829 1364 1406 1532">▪ Quando o fim de curso inferior FCI e o fim de curso superior FCS são acionados, os fins de curso passam de azul para amarelo.

Figura 3.12: Reposta visual do sensor AF e dos fins de curso (FCI e FCS).

Quando se passa com o cursor do rato por cima dos botões, estes ficam azuis (Figuras 3.13, 3.16 e 3.18).

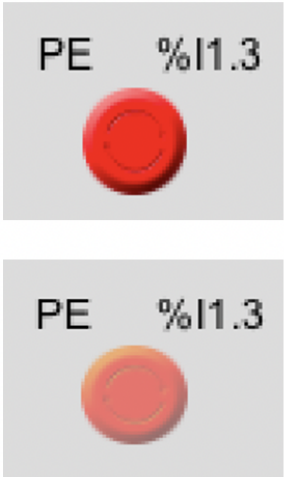
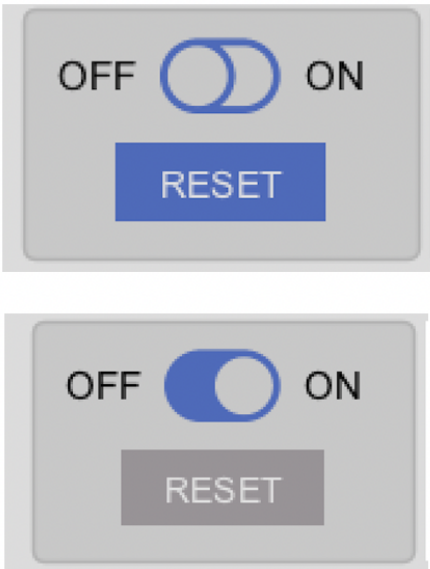
VISUALIZAÇÃO	ESTADO
	<p>PARAGEM DE EMERGÊNCIA (PE) %I1.3</p> <ul style="list-style-type: none"> Quando o botão de PE é pressionado, o botão passa da cor normal, vermelho vivo, para um vermelho esbranquiçado; Quando o sistema de PE deixa de estar ativo, o botão passa de vermelho esbranquiçado para o vermelho normal.
	<p>OFF/ON e RESET</p> <ul style="list-style-type: none"> Botão tipo “<i>slider</i>” OFF >> ON Botão OFF/ON na cor cinza, o simulador está desligado; Botão OFF/ON na cor azul, o simulador está ligado.

Figura 3.13: Reposta visual dos botões PE e OFF/ON.

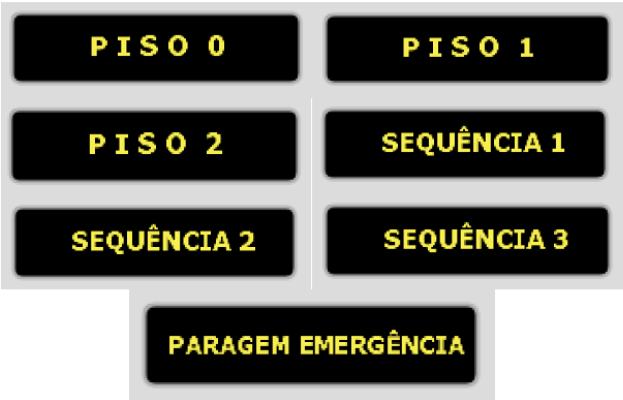
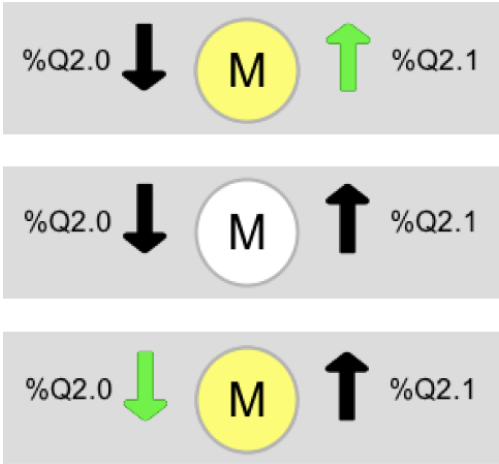
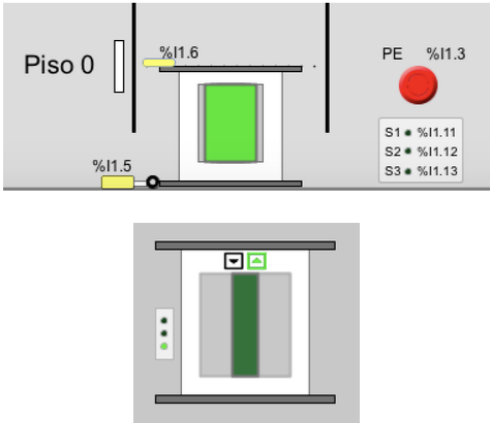
VISUALIZAÇÃO	ESTADO
	<p>CAIXA DE INFORMAÇÕES</p> <ul style="list-style-type: none"> Permite saber o estado do elevador no momento.
	<p>SETAS SUBIR/DESCER MOTOR</p> <ul style="list-style-type: none"> Seta de subir a verde, significa que o relé de subida (KM1) recebeu ordem; Seta de descer a verde, significa que o relé de descida (KM0) recebeu ordem; O símbolo do motor (M) fica amarelo quando o elevador está em movimento.
	<p>Estado do Elevador</p> <ul style="list-style-type: none"> Ao lado da estrutura do elevador encontra-se o botão PE e um bloco onde é possível visualizar a sequência selecionada; Na imagem da cabine, é possível visualizar o estado do elevador no momento (P0, P1, P2).

Figura 3.14: Visualização do estado do simulador.

3.3.3 Interface Gráfica com o Utilizador (GUI)

Na Figura 3.15 é apresentada a interface gráfica do simulador.

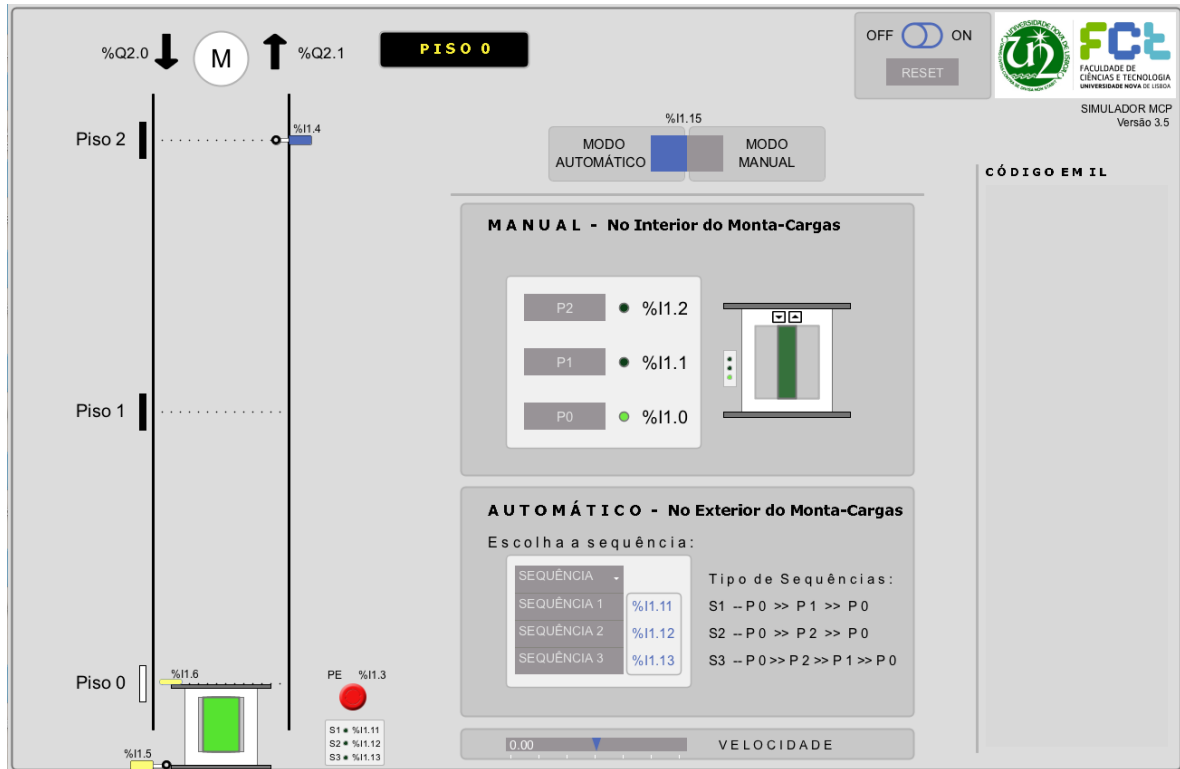


Figura 3.15: Interface gráfica com o utilizador (GUI) em linguagem *Processing*.

Observando a Figura 3.15, no lado esquerdo é apresentado o modelo gráfico de elevador, uma estrutura em 2D, dividida em 3 pisos - Piso 0, Piso 1 e Piso 2.

Inicialmente, o elevador encontra-se posicionado no piso 0.

Na interface gráfica do simulador (Figura 3.15) estão presentes os endereços de todas as variáveis que têm influência no funcionamento do elevador.

Na estrutura do elevador, do lado esquerdo, estão distribuídos 3 ímanes ao longo da guia, um por cada piso. No interior da estrutura fica a cabina do elevador, composta pelo sensor AF. O sensor AF (%I1.6) está posicionado no lado esquerdo, no topo da cabina. Este sensor permite detetar a posição actual da cabina ao longo do curso.

No piso 0, do lado esquerdo, junto à base da cabina, encontra-se o fim de curso inferior (FCI - %I1.5), tendo como função detetar se a cabina se encontra no piso 0, ou abaixo deste, no caso da cabina exceder o seu curso normal. No piso 2, do lado direito, encontra-se o fim de curso superior (FCS - %I1.4) que permite detetar se a cabina se encontra no piso 2, ou acima deste.

Junto à estrutura do elevador está localizado o botão de paragem de emergência (PE - %I1.3). No caso deste botão ser pressionado, a cabina pára e, após 4 segundos, é enviada para o piso 0 (piso inicial).

Ao lado da estrutura do elevador, encontra-se o conjunto de botões que permitem controlar o elevador: modo manual e modo automático.

Modo Manual

Como referido anteriormente, na secção 3.3.1.1, o modo manual é composto por 3 botões - P0, P1 e P2.

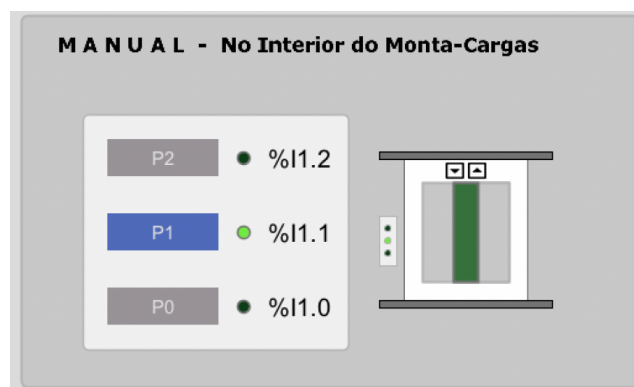


Figura 3.16: Modo manual na interface gráfica do simulador. Elevador parado no piso 1.

Na interface gráfica do simulador, no modo manual (Figura 3.16) é possível visualizar o piso onde o elevador se encontra e ainda verificar quando este recebe ordem para subir ou descer de piso. Num sistema real, este modo poderia estar no interior da cabina do elevador, como já foi referido anteriormente.

Colocando o simulador em modo manual é possível controlar o elevador caso exista alguma falha ou avaria do sistema de controlo automático.

Num sistema real, este modo só deveria ser utilizado pelo pessoal responsável pela manutenção do elevador.

Modo Automático

O modo automático, como referido na secção 3.3.1.1, é composto por um bloco selector, sendo possível escolher entre 3 sequências - Sequência 1, Sequência 2 e Sequência 3.

Na interface gráfica do simulador, no modo automático (Figura 3.17) é possível visualizar os tipos de sequência disponíveis.

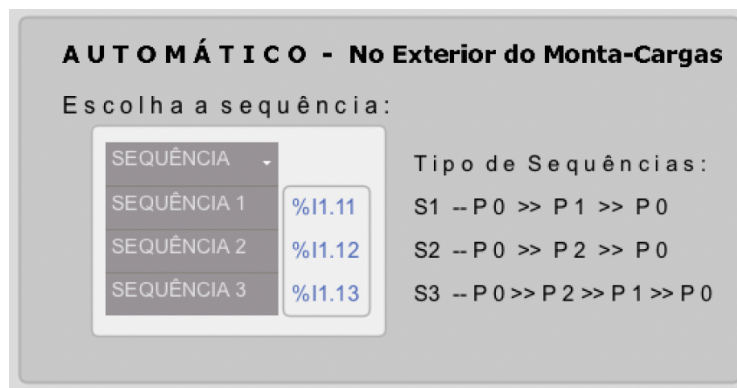


Figura 3.17: Modo automático na interface gráfica do simulador.

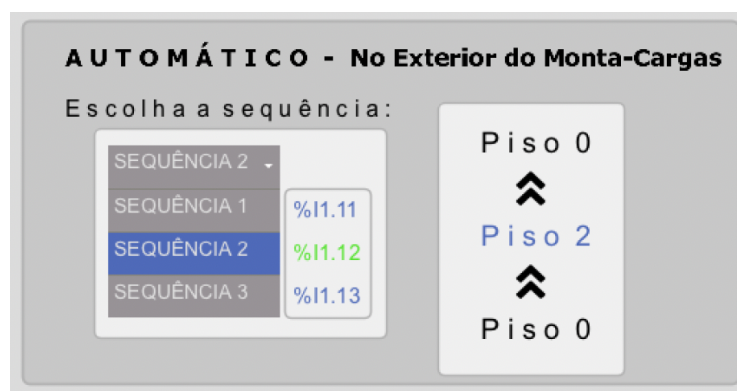


Figura 3.18: Modo automático na interface gráfica, quando é selecionada a sequência 2.

Quando se seleciona uma sequência, por exemplo, a sequência 2 (Figura 3.18), o endereço do botão sequência 2 fica a verde (%I1.12) e o tipo de sequência é apresentada à direita (P0 » P2 » P0).

No piso 0, junto à estrutura do elevador, é possível visualizar um bloco com a sequência selecionada (Figura 3.14 - Estado do Elevador).

Num sistema real, este modo poderia estar localizado junto à estrutura do elevador.

Também na interface gráfica é possível visualizar o código em linguagem IL, referente às sequências de funcionamento automático e assim ir acompanhando o movimento do elevador. Também é possível visualizar o código associado ao botão P1 do modo manual e ao botão de paragem de emergência (PE).

3.3.3.1 Código em Lista de Instruções

Conforme introduzido na secção 2.2.3, a linguagem em lista de instruções (IL) é uma linguagem de baixo nível utilizada na programação de microprocessadores (similar ao *Assembly*), fazendo parte do padrão IEC 61131-3. Tem regras de semântica muito bem definidas, que são utilizadas nas instruções.

Na interface gráfica do simulador é apresentado o código em linguagem lista de instruções associado ao botão de paragem de emergência e aos modos de funcionamento do elevador.

O simulador pode ser utilizado como recurso didático para o ensino de linguagens de programação. O utilizador pode facilmente traduzir o código em lista de instruções (IL), presente no simulador, para a linguagem *Ladder* (LD).

O código desenvolvido, em linguagem lista de instruções (IL), não é editável e não é interpretado, pois apenas pode ser visualizado na interface do simulador.

Na interface gráfica do simulador (Figura 3.15) estão presentes os endereços de todas as variáveis que têm influência no funcionamento do elevador.

Como referido anteriormente, o botão de paragem de emergência (PE - %I1.3), presente na interface do simulador, quando pressionado, imobiliza a cabina e envia o elevador para o piso 0. Ao botão PE está associado um código em IL.

O modo manual é composto por três botões - P0, P1 e P2:

- **Botão P0** - %I1.0 – envia a cabina para piso 0;
- **Botão P1** - %I1.1 – envia a cabina para piso 1;
- **Botão P2** - %I1.2 – envia a cabina para piso 2.

Quando é pressionado o botão P1, o código em IL correspondente aparece do lado direito na interface do simulador.

Para o modo automático foram definidas 3 sequências de funcionamento do elevador, onde:

- **Sequência 1** - %I1.11 – Piso 0 » Piso 1 » Piso 0;
- **Sequência 2** - %I1.12 – Piso 0 » Piso 2 » Piso 0;
- **Sequência 3** - %I1.13 – Piso 0 » Piso 2 » Piso 1 » Piso 0.

A cada sequência está associado um código em linguagem IL.

Quando é selecionada uma sequência, o código IL correspondente a essa sequência, aparece do lado direito na interface do simulador.

Na tabela 3.1 são apresentados os endereços de todas as entradas e saídas utilizadas no simulador.

Tabela 3.1: Lista de endereços das entradas/saídas utilizadas no simulador.

Descrição	Sigla	Endereço
Botão Piso 0	P0	%I1.0
Botão Piso 1	P1	%I1.1
Botão Piso 2	P2	%I1.2
Botão Paragem de Emergência	PE	%I1.3
Fim de Curso Superior	FCS	%I1.4
Fim de Curso Inferior	FCI	%I1.5
Sensor Ampola Frente	AF	%I1.6
Botão Sequência 1	SEQ1	%I1.11
Botão Sequência 2	SEQ2	%I1.12
Botão Sequência 3	SEQ3	%I1.13
Botão Automático/Manual	AUTO/MAN	%I1.15
Relé Descida	KM0	%Q2.0
Relé Subida	KM1	%Q2.1
Memória Piso 0	-	%M0
Memória Piso 1	-	%M1
Memória Piso 2	-	%M2
<i>Timer</i> Piso 0	TM0	%TM0.Q
<i>Timer</i> Piso 1	TM1	%TM1.Q
<i>Timer</i> Piso 2	TM2	%TM2.Q
<i>Bit</i> do Sistema	-	%S13

Em seguida, são apresentados os códigos em IL associados ao botão de paragem de emergência (PE), ao botão P1 do modo manual e às sequências de funcionamento do modo automático.

O primeiro passo na elaboração de um programa é normalmente a criação de um modelo que descreva as especificações funcionais do sistema, como é o caso do *Grafcet*.

Tendo o *Grafcet* torna-se mais fácil traduzir para outras linguagens de programação, que fazem parte do padrão IEC 61131-3, como a linguagem *Ladder* e a linguagem SFC.

Deste modo é apresentado um exemplo de um *Grafcet* que traduz o funcionamento do botão de paragem de emergência (PE) (figura 3.19).

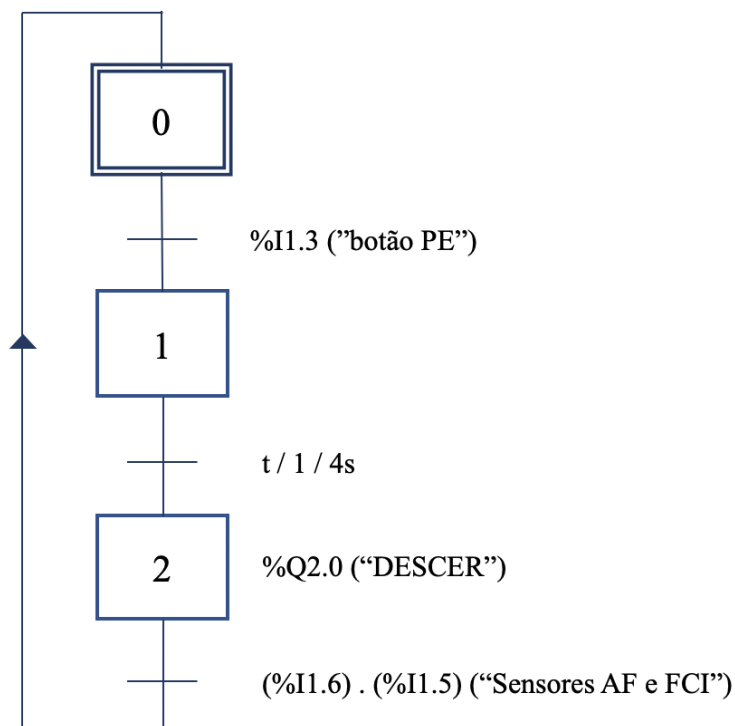


Figura 3.19: Grafcet - Botão de paragem de emergência (PE).

A partir do *Grafcet* foi feita a conversão para a linguagem lista de instruções (IL).

O código desenvolvido em IL traduz o funcionamento do botão de paragem de emergência (PE).

Paragem de Emergência – PE » P0

Listagem 3.1: Paragem de Emergência

```

1  ! ( * PE * )
2
3  LD      %I1.3      ! ( * Botao PE pressionado * )
4  S       %M7
5
6  LD      %TM1.Q
7  ST      %Q2.0
8
9  LD      %I1.6
10 AND     %I1.5
11 R       %M7

```

Quando é pressionado o botão PE é feito *set* à memória %M7 que dá ordem ao *timer* TM1. Como o *timer* TM1 é do tipo Ton, após 4 segundos, o elevador recebe ordem de descida (%Q2.0) para o piso 0. Quando o sensor AF (%I1.6) deteta a presença do elevador no piso 0 e o fim de curso FCI (%I1.5) é atuado, é feito *reset* à memória %M7.

Listagem 3.2: Paragem de Emergência - *Timer*

```

1  ! ( * Ton %TM1: PE , 4s * )
2  BLK      %TM1
3  LD      %M7
4  IN
5  END_BLK

```

Da mesma forma, foi desenvolvido um código em IL referente ao botão P1 do modo manual, que permite enviar o elevador do piso 0 para piso 1.

Botão P1 – P0 » P1

Listagem 3.3: Botão P1

```

1  !(* P0 -> P1 *)
2
3  LD          %S13
4  AND         %I1.6
5  AND         %I1.5
6  ANDN        %I1.15
7  S           %M0          !(* Elevador no piso 0 *)
8
9  LD          %M0
10 AND         %I1.1        !(* Botao P1 pressionado *)
11 ANDN        %I1.0
12 ANDN        %I1.2
13 S           %M8
14
15 LD          %TM0.Q
16 ST          %Q2.1
17
18 LDR         %I1.6
19 AND         %M0
20 AND         %Q2.1
21 R           %M0
22 R           %M8

```

Assim que o programa é iniciado, é verificado se o elevador se encontra no piso 0. Caso não esteja, este é enviado para o piso 0.

Quando é pressionado o botão P1 é feito *set* à memória %M8 que dá ordem ao *timer* TM0. Como o *timer* TM0 é do tipo Ton, após 4 segundos, o elevador recebe ordem de subida (%Q2.1) para o piso 1. Quando o sensor AF (%I1.6) deteta a presença do elevador no piso 1, é feito *reset* a todas as memórias.

Listagem 3.4: Botão P1 - Timer

```

1  !(* Ton %TM0: P0, 4s *)
2  BLK         %TM0
3  LD          %M8
4  IN
5  END_BLK

```


Foram também desenvolvidos três códigos em linguagem lista de instruções (IL) para cada sequência de funcionamento do modo automático, que são apresentados em seguida.

O utilizador pode acompanhar o movimento do elevador a partir do código IL.

Sequência 1 – P0 » P1 » P0

Listagem 3.5: Sequência 1

```

1  !(* SEQ 1: P0 -> P1 -> P0 *)
2
3  LD      %S13
4  AND     %I1.15
5  ANDN    %I1.5
6  OR      %TM1.Q
7  ST      %Q2.0
8
9  LD      %I1.6
10 AND     %I1.5
11 S       %M0      !(* Elevador no piso 0 *)
12
13 LD      %M0
14 AND     %I1.11    !(* Sequência 1 seleccionada *)
15 ANDN    %I1.12
16 ANDN    %I1.13
17 S       %M8
18
19 LD      %TM0.Q
20 ST      %Q2.1
21
22 LDR     %I1.6
23 AND     %M0
24 AND     %Q2.1
25 S       %M1      !(* Elevador no piso 1 *)
26 R       %M0
27 R       %M8
28 S       %M7
29
30 LD      %I1.6
31 AND     %I1.5
32 AND     %M1
33 AND     %Q2.0
34 R       %M1
35 R       %M7

```

Assim que o programa é iniciado, é verificado se o elevador se encontra no piso 0. Caso não esteja, este é enviado para o piso 0.

Quando é seleccionada a sequência 1 é feito *set* à memória %M8 que dá ordem ao *timer* TM0. Como o *timer* TM0 é do tipo Ton, após 4 segundos, o elevador recebe ordem de subida (%Q2.1) para o piso 1.

Quando o sensor AF (%I1.6) deteta a presença do elevador no piso 1, é feito *reset* à memória %M8 do *timer* TM0 e *set* à memória %M7, que dá ordem ao *timer* TM1. Como o *timer* TM1 é do tipo Ton, após 4 segundos, o elevador recebe ordem de descida (%Q2.0) para o piso 0.

Por fim, é verificado se elevador se encontra no piso 0 e feito *reset* a todas as memórias.

Listagem 3.6: Sequência 1 - *Timers*

```
1  !(* Ton %TM1: P1, 4s *)
2  BLK          %TM1
3  LD          %M7
4  IN
5  END_BLK
6
7  !(* Ton %TM0: P0, 4s *)
8  BLK          %TM0
9  LD          %M8
10 IN
11 END_BLK
```

Sequência 2 – P0 » P2 » P0

Listagem 3.7: Sequência 2

```

1  !(* SEQ 2: P0 -> P2 -> P0 *)
2
3  LD          %S13
4  AND         %I1.15
5  ANDN        %I1.5
6  OR          %TM2.Q
7  ST          %Q2.0
8
9  LD          %I1.6
10 AND         %I1.5
11 S           %M0          !(* Elevador no piso 0 *)
12 R           %M2
13 R           %M9
14
15 LD          %I1.6
16 AND         %I1.4
17 S           %M2          !(* Elevador no piso 2 *)
18 R           %M0
19 R           %M8
20 S           %M9
21
22 LD          %M0
23 AND         %I1.12        !(* Sequência 2 selecionada *)
24 ANDN        %I1.11
25 ANDN        %I1.13
26 S           %M8
27
28 LD          %TM0.Q
29 ST          %Q2.1

```

Assim que o programa é iniciado, é verificado se o elevador se encontra no piso 0. Caso não esteja, este é enviado para o piso 0.

Quando é selecionada a sequência 2 é feito *set* à memória %M8 que dá ordem ao *timer* TM0. Como o *timer* TM0 é do tipo Ton, após 4 segundos, o elevador recebe ordem de subida (%Q2.1) para o piso 2.

Quando o sensor AF (%I1.6) deteta a presença do elevador no piso 2 e o fim de curso FCS (%I1.4) é atuado, é feito *reset* à memória %M8 do *timer* TM0 e *set* à memória %M9, que dá ordem ao *timer* TM2. Como o *timer* TM2 é do tipo Ton, após 4 segundos, o elevador recebe ordem de descida (%Q2.0) para o piso 0.

Por fim, é verificado se elevador se encontra no piso 0 e feito *reset* a todas as memórias.

Listagem 3.8: Sequência 2 - *Timers*

```
1  !(* Ton %TM2: P2, 4s *)
2  BLK          %TM2
3  LD           %M9
4  IN
5  END_BLK
6
7  !(* Ton %TM0: P0, 4s *)
8  BLK          %TM0
9  LD           %M8
10 IN
11 END_BLK
```

Sequência 3 – P0 » P2 » P1 » P0

Listagem 3.9: Sequência 3

```

1  !(* SEQ 3: P0 -> P2 -> P1 -> P0 *)
2
3  LD      %S13
4  AND     %I1.15
5  ANDN    %I1.5
6  OR      %TM2.Q
7  OR      %TM1.Q
8  ST      %Q2.0
9
10 LD      %I1.6
11 AND     %I1.5
12 S       %M0          !(* Elevador no piso 0 *)
13
14 LD      %M0
15 AND     %I1.13        !(* Sequência 3 selecionada *)
16 ANDN    %I1.11
17 ANDN    %I1.12
18 S       %M8
19
20 LD      %TM0.Q
21 ST      %Q2.1
22
23 LD      %I1.6
24 AND     %I1.4
25 S       %M2          !(* Elevador no piso 2 *)
26 R       %M0
27 R       %M8
28 S       %M9
29
30 LDR     %I1.6
31 AND     %M2
32 AND     %Q2.0
33 S       %M1          !(* Elevador no piso 1 *)
34 R       %M2
35 R       %M9
36 S       %M7
37
38 LD      %I1.6
39 AND     %I1.5
40 AND     %M7
41 AND     %Q2.0
42 R       %M1
43 R       %M7

```

Assim que o programa é iniciado, é verificado se o elevador se encontra no piso 0. Caso não esteja, este é enviado para o piso 0.

Quando é selecionada a sequência 3 é feito *set* à memória %M8 que dá ordem ao *timer* TM0. Como o *timer* TM0 é do tipo Ton, após 4 segundos, o elevador recebe ordem de subida (%Q2.1) para o piso 2.

Quando o sensor AF (%I1.6) deteta a presença do elevador no piso 2 e o fim de curso FCS (%I1.4) é atuado, é feito *reset* à memória %M8 do *timer* TM0 e *set* à memória %M9, que dá ordem ao *timer* TM2. Como o *timer* TM2 é do tipo Ton, após 4 segundos, o elevador recebe ordem de descida (%Q2.0) para o piso 1.

Quando o sensor AF (%I1.6) deteta a presença do elevador no piso 1, é feito *reset* à memória %M9 do *timer* TM2 e *set* à memória %M7, que dá ordem ao *timer* TM1. Como o *timer* TM1 é do tipo Ton, após 4 segundos, o elevador recebe ordem de descida (%Q2.0) para o piso 0.

Por fim, é verificado se elevador se encontra no piso 0 e feito *reset* a todas as memórias.

Listagem 3.10: Sequência 3 - *Timers*

```
1  !(* Ton %TM2: P2, 4s *)
2  BLK          %TM2
3  LD          %M9
4  IN
5  END_BLK
6
7  !(* Ton %TM1: P1, 4s *)
8  BLK          %TM1
9  LD          %M7
10 IN
11 END_BLK
12
13 !(* Ton %TM0: P0, 4s *)
14 BLK          %TM0
15 LD          %M8
16 IN
17 END_BLK
```

RESULTADOS EXPERIMENTAIS

No presente capítulo são apresentados os modos de funcionamento do elevador na interface gráfica com o utilizador (GUI).

Na secção [4.1.1](#) é ilustrado o estado do elevador quando é seleccionado o modo de funcionamento manual.

Na secção [4.1.2](#) é apresentado o estado do elevador quando o modo funcionamento automático está activo.

O estado do elevador no momento em que é pressionado o botão de paragem de emergência é ilustrado na secção [4.1.3](#).

4.1 Modos de funcionamento do Simulador

Como foi referido anteriormente, o simulador pode ser controlado através de dois modos de funcionamento: modo manual e modo automático.

Em seguida serão apresentados os modos de funcionamento do simulador do tipo elevador monta-cargas.

A Figura 4.1 ilustra o estado inicial do simulador.

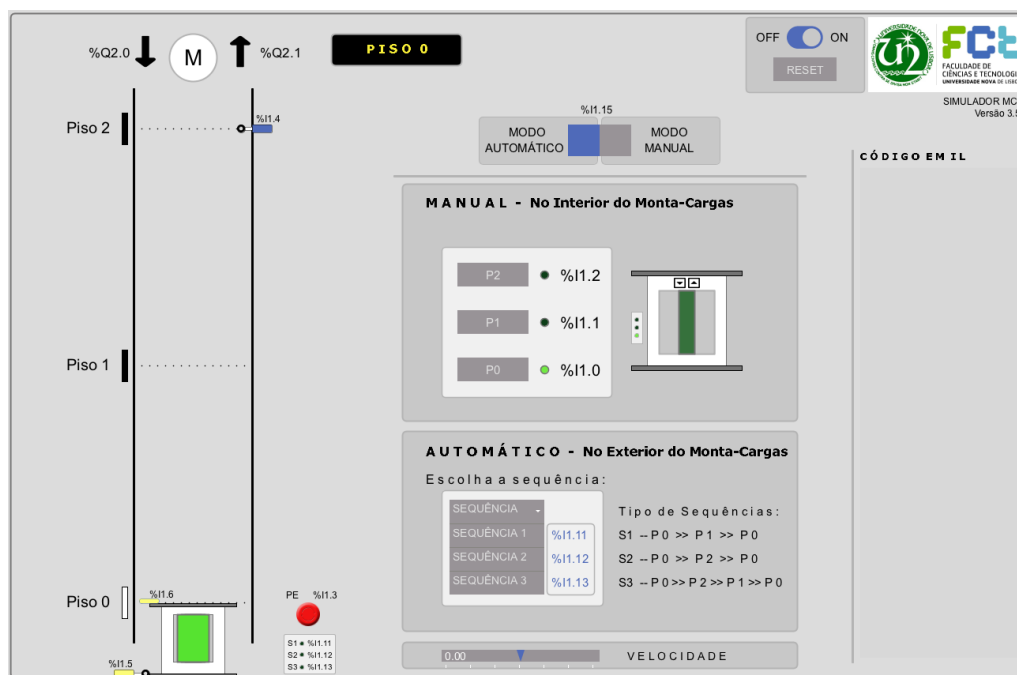


Figura 4.1: Estado inicial da interface gráfica do simulador.

No estado inicial, a cabina do elevador encontra-se no piso 0 e com o modo automático selecionado.

O Botão OFF/ON já está ligado (ON).

Neste momento o elevador está pronto para receber uma ordem.

4.1.1 Modo de Funcionamento Manual

Para o elevador funcionar em modo manual é necessário selecionar no botão automático/-manual o modo pretendido.

Botão Piso 1

Para enviar a cabina para o piso 1, basta pressionar o botão P1. A cabina só sai do piso 0 após 4 segundos.

Na Figura 4.2 está representado o estado do simulador durante a transição Piso 0 » Piso 1.

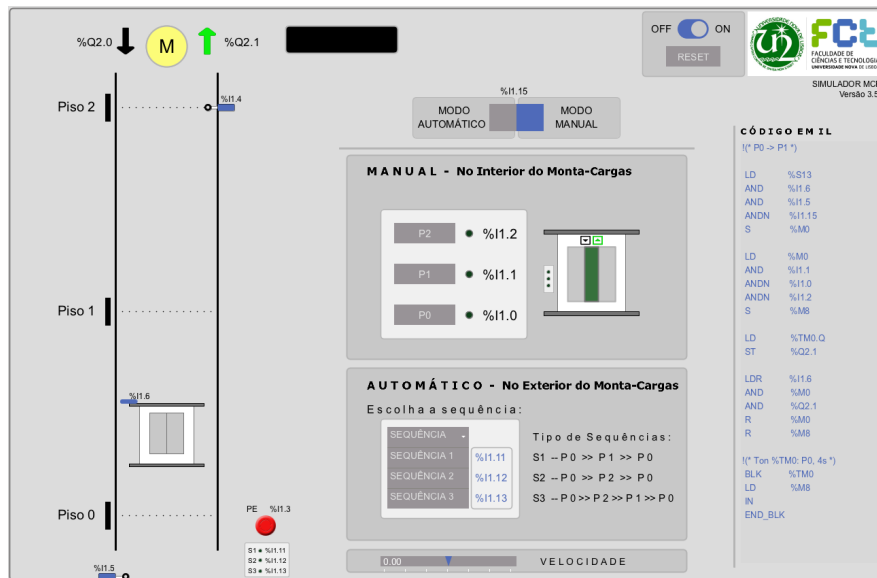


Figura 4.2: Interface gráfica do simulador na transição Piso 0 » Piso 1.

Como se pode observar na Figura 4.2, o sensor AF deixa de detetar o íman. A cor do sensor AF passa de amarelo para azul, pois o sensor AF deixa de estar sob o efeito do campo magnético do íman. O íman também muda de cor (de branco para preto).

Também o fim de curso inferior (FCI) deixa de estar actuado e muda de estado (passa de amarelo para azul).

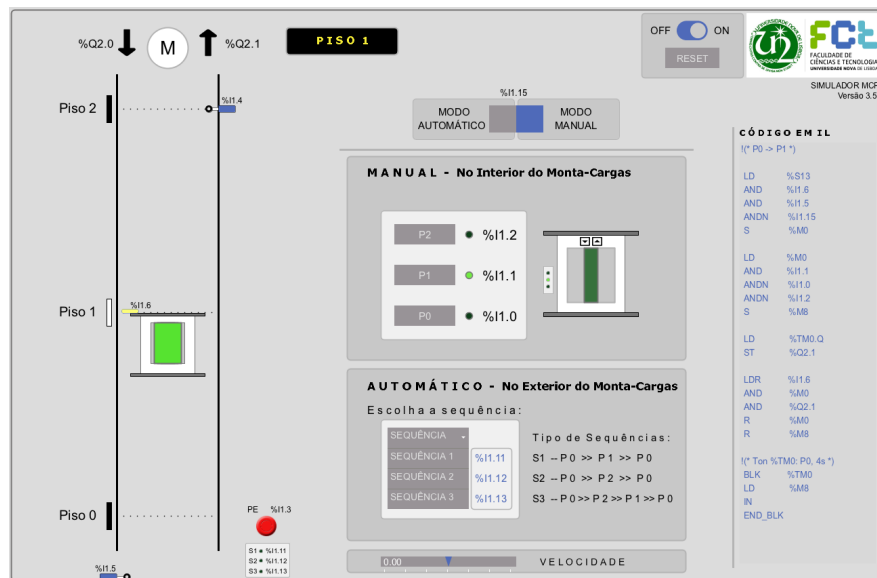


Figura 4.3: Modo manual - Elevador parado no Piso 1.

Ao observar a Figura 4.3, verifica-se que a cabina se encontra no piso 1.

Na caixa de informações, situada no topo da interface gráfica do simulador, está presente a informação do piso actual.

O sensor AF detetou o íman. A cor do sensor AF passou de azul para amarelo e a do

íman passou de preto para branco.

Do lado direito na interface do simulador é apresentado o código em IL correspondente ao botão P1 (Figura 4.2 e 4.3).

Botão Piso 2

Para enviar a cabina para o piso 2 basta carregar no botão P2.

A cabina pode sair do piso 1 directamente para o piso 2.

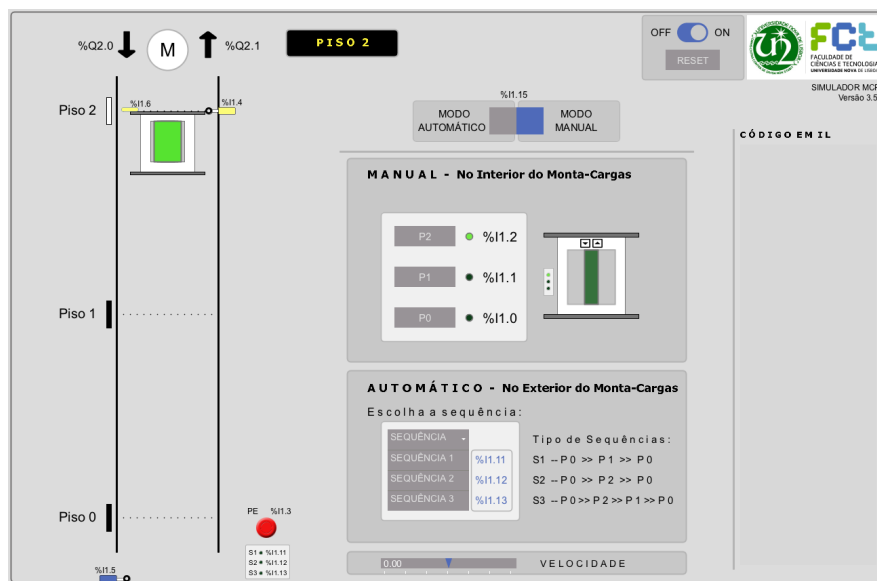
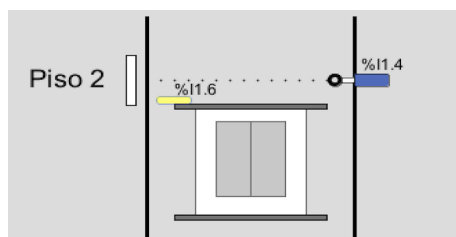
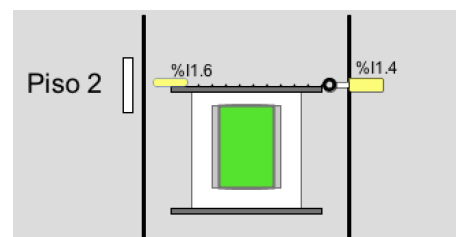


Figura 4.4: Modo manual - Elevador parado no Piso 2.

Como se pode observar na Figura 4.4, a cabina encontra-se no piso 2.



(a) Sensor AF deteta o íman do piso 2



(b) Abertura das portas do elevador e mudança de estado do FCS

Figura 4.5: Chegada da cabina ao piso 2.

Na Figura 4.5 pode-se observar quando o sensor AF deteta o íman no piso 2 (Figura 4.5(a)) e a abertura das portas do elevador quando este chega ao piso 2 (Figura 4.5(b)).

A cabina ao chegar ao piso 2 faz actuar o fim de curso superior (FCS) (passa de azul para amarelo) (Figura 4.5).

4.1.2 Modo de Funcionamento Automático

Para definir este modo o botão automático/manual tem de estar selecionado no modo automático.

No modo automático pode-se escolher entre 3 sequências de funcionamento diferentes, nomeadamente:

- Sequência 1 – Piso 0 » Piso 1 » Piso 0;
- Sequência 2 – Piso 0 » Piso 2 » Piso 0;
- Sequência 3 – Piso 0 » Piso 2 » Piso 1 » Piso 0.

Botão de Sequência 1

Na Figura 4.6 está representado o estado do simulador durante a transição da sequência 1 - Piso 0 » Piso 1.

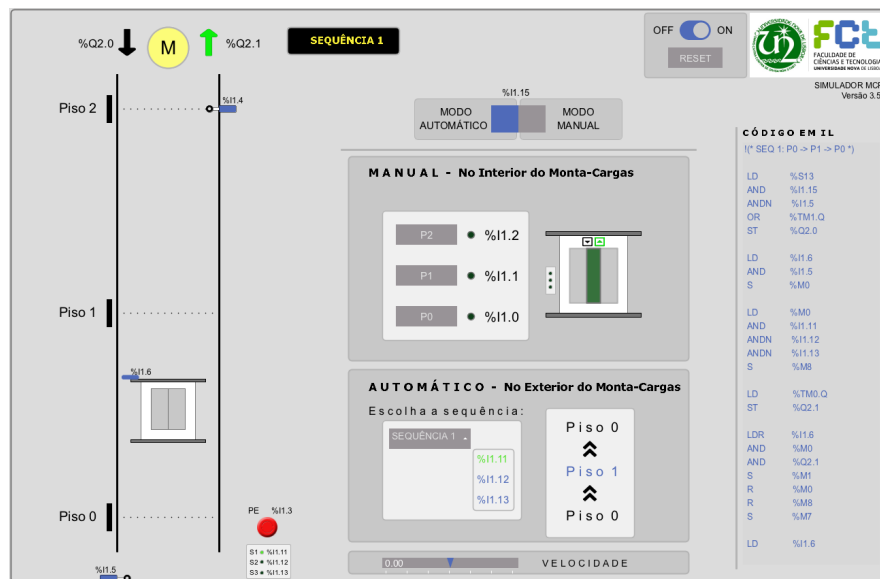


Figura 4.6: Interface gráfica do simulador na transição da sequência 1 - Piso 0 » Piso 1.

No lado direito da interface gráfica do simulador é apresentado o código em IL correspondente à sequência 1. Quando o elevador recebe ordem de subida, verifica-se que a seta correspondente ao relé de subida (%Q2.1) muda da cor preto para verde.

Como se pode observar na Figura 4.6, a sequência selecionada aparece na caixa de informações e também na caixa de botões junto ao elevador.

Após a cabina chegar ao piso 1, espera 4 segundos nesse piso antes de partir para o piso 0.

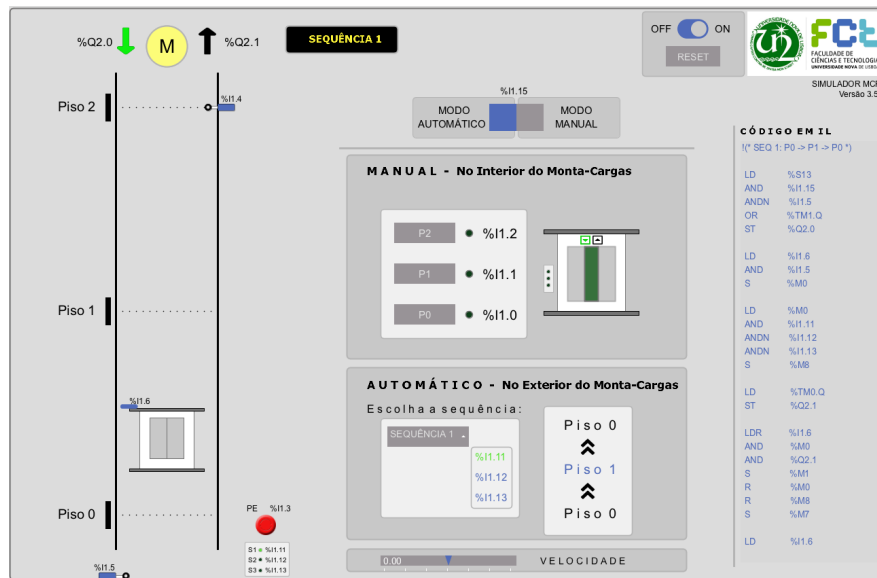


Figura 4.7: Interface gráfica do simulador na transição da sequência 1 - Piso 1 » Piso 0.

Como se pode observar na Figura 4.7, quando o elevador recebe ordem para descer, verifica-se que a seta correspondente ao relé de descida (%Q2.0) muda da cor preto para verde.

No final da sequência, o elevador fica pronto para receber uma nova ordem.

Botão de Sequência 2

Na Figura 4.8 pode-se verificar que a sequência selecionada foi a sequência 2 (P0 » P2 » P0).

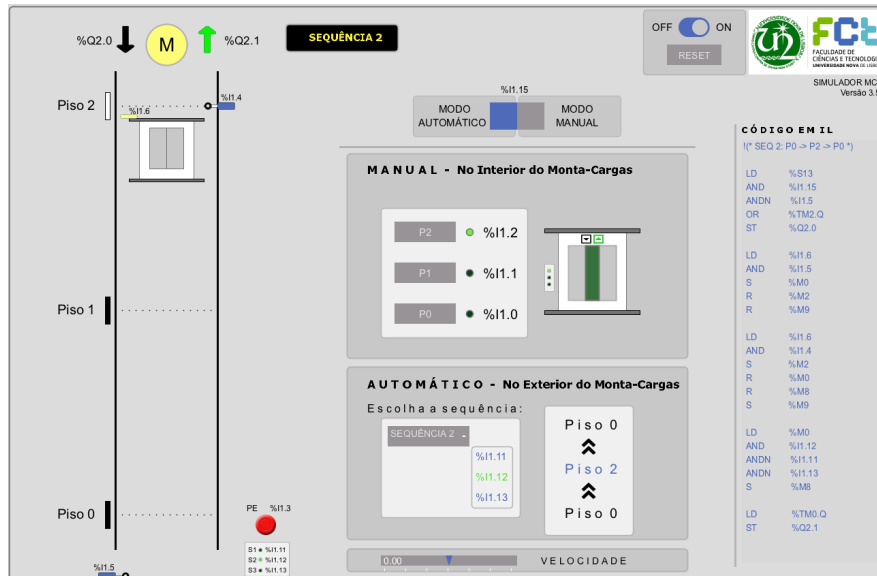


Figura 4.8: Interface gráfica do simulador na transição da sequência 2 - Piso 0 » Piso 2.

O funcionamento da sequência 2 é semelhante ao funcionamento da sequência 1.

Após a chegada ao piso 2, a cabina recebe ordem para descer para o piso 0.

No lado direito do simulador é apresentado o código em IL correspondente à sequência 2.

Botão de Sequência 3

Observando a Figura 4.9 verifica-se que a sequência 3 foi a sequência selecionada (P0 » P2 » P1 » P0).

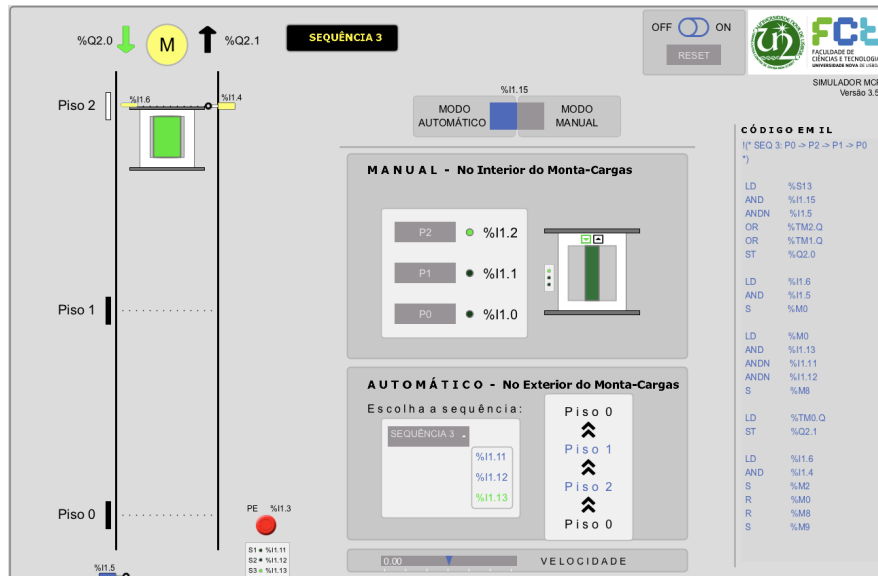


Figura 4.9: Cabina pronta para começar a descer, na transição da sequência 3 - P2 » P1.

Na Figura 4.9 verifica-se o momento em que a cabina está prestes a sair do piso 2 para o piso 1, estando já presente a ordem de descida do relé (%Q2.0).

No lado direito do simulador é apresentado o código em IL correspondente à sequência 3.

Na Figura 4.10, a cabina do elevador encontra-se no piso 1 e, após 4 segundos, irá descer para o piso 0.

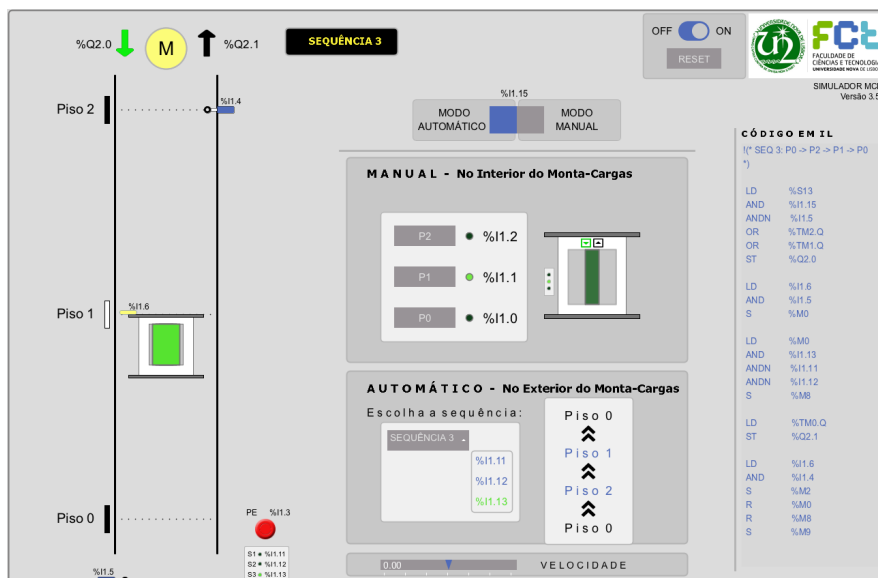


Figura 4.10: Interface gráfica do simulador na sequência 3 - Elevador parado no Piso 1.

Na Figura 4.11 está representada a chegada da cabina ao piso 0, chegando ao fim da sequência 3.

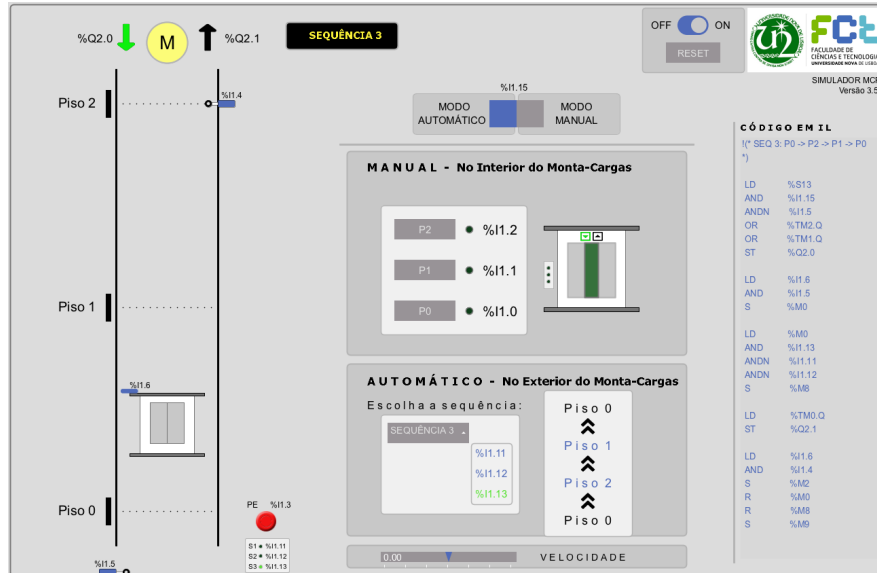


Figura 4.11: Interface gráfica do simulador no final da sequência 3, na transição P1 » P0.

4.1.3 Paragem de Emergência

Caso ocorra uma situação de emergência, como uma avaria ou falha do sistema, pode ser necessária a paragem do elevador nesse instante.

Para parar o elevador em segurança basta carregar no botão de paragem de emergência (PE - Botão Vermelho - %I1.3). Como referido na secção 3.3.1.1, caso o botão seja premido, a cabina pára e, após 4 segundos, é enviada para o piso 0 (piso inicial).

Na Figura 4.12 é apresentado o caso em que a cabina se desloca para o piso 2, em modo automático (sequência 2).

Foi carregado no botão PE e a cabina parou nesse instante.

O elevador recebe ordem para descer (%Q2.0) e, após uma espera de 4 segundos, a cabina desce para o piso inicial.

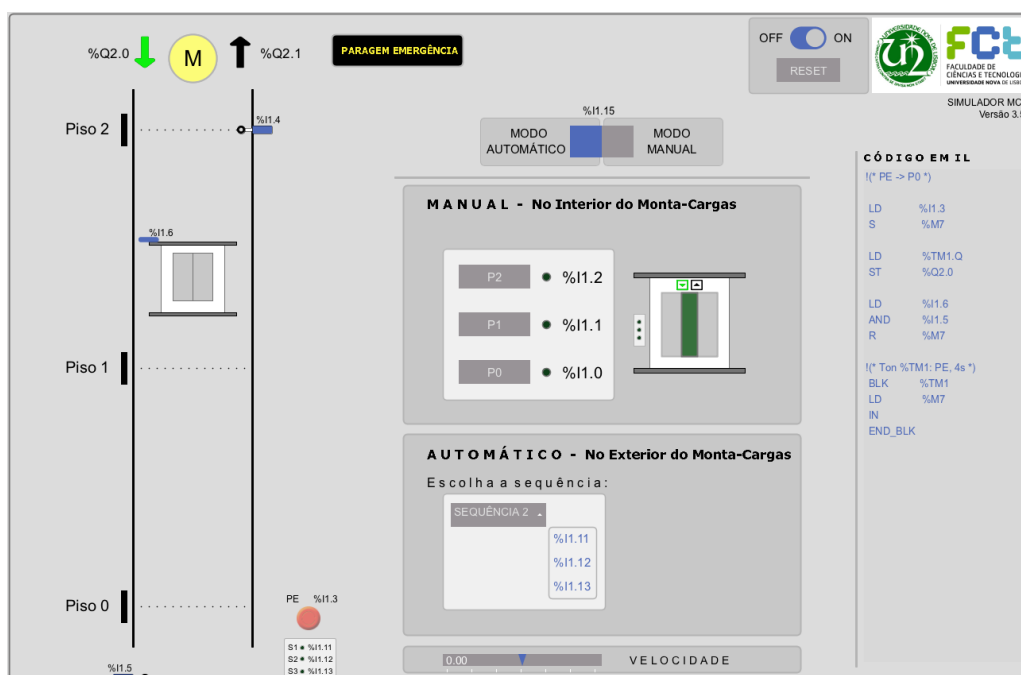


Figura 4.12: A paragem de emergência (PE) é acionada.

Na caixa de informações, no topo do simulador, está presente a mensagem “Paragem Emergência” (Figura 4.12).

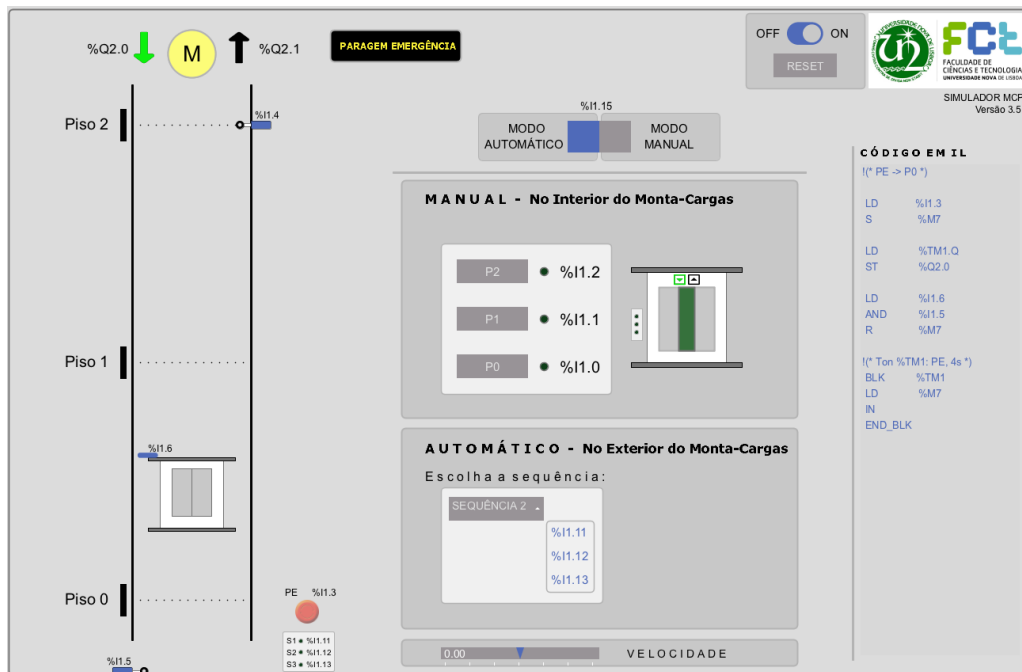


Figura 4.13: Após ter sido acionado o botão de paragem de emergência, o elevador desce para o piso 0.

Na Figura 4.13 pode-se observar a chegada da cabina ao piso 0, após ter sido pressionado o botão PE.

O botão PE mantém-se activo enquanto a cabina não chega ao piso 0 (botão PE com cor vermelho esbranquiçado). Quando a cabina chega ao piso 0, este deixa de estar activo (botão PE passa a vermelho vivo) (figura ??).

No lado direito do simulador é apresentado o código em IL correspondente ao botão PE (Figura 4.12 e 4.13).

4.2 Validação do Simulador

Os objectivos propostos para a interface gráfica do simulador foram cumpridos com sucesso.

- **Modos de funcionamento:**

- Modo Manual – A partir do modo manual é possível controlar o elevador permitindo ao utilizador seleccionar apenas o piso pretendido;
- Modo Automático – A partir do modo automático o utilizador pode escolher a sequência de funcionamento que pretende para o elevador.

→ Foram atingidos os objectivos requeridos para os modos de funcionamento do elevador.

- **Visualização do código em linguagem lista de instruções (IL)**

Na interface gráfica do simulador é possível acompanhar, a partir do código em lista de instruções (IL), o modo de funcionamento seleccionado.

→ O simulador é facilmente adaptável a outras linguagens de programação. Neste documento é apresentado o *Grafcet* para o botão PE, ficando como exercício didático criar o *Grafcet* para os outros botões, a partir dos códigos IL existentes. Além disso, o utilizador poderá ainda criar o código em linguagem IL para os botões P0 e P2 do modo manual.

Tendo sido verificados e satisfeitos os requisitos estabelecidos para o simulador, conclui-se que o simulador vai resultar numa vantagem para os alunos de modo a consolidarem conhecimentos sobre linguagens de programação, assim como, ajudar a perceber quais os parâmetros a ter em consideração aquando do controlo do elevador do tipo monta-cargas.

CONCLUSÕES E TRABALHO FUTURO

No capítulo final é apresentado um resumo dos pontos mais importantes no desenvolvimento do simulador, destacando-se a importância da realização do simulador para o ensino da linguagem de programação lista de instruções (IL).

São apresentadas algumas propostas para desenvolver em trabalho futuro que possam vir a contribuir para uma melhoria contínua do desempenho do simulador.

5.1 Conclusões

O desenvolvimento do simulador e a respectiva interface gráfica com o utilizador foi construída com sucesso, baseando-se num conjunto de melhorias e testes que foram feitos ao longo de todo o trabalho.

Um aspecto importante que importa referir foi a constante melhoria do simulador para que no final o elevador monta-cargas funcionasse de forma adequada, nos termos das especificações definidas previamente, semelhantes às usadas em sistemas reais.

A partir do simulador desenvolvido em linguagem *Processing* é possível testar as funcionalidades típicas de um elevador monta-cargas:

- Controlar o elevador em modo manual. Para o modo manual foram criados 3 botões com a função de dar ordem à cabina para se deslocar entre os 3 pisos existentes (Piso 0, Piso 1 e Piso 2). O modo manual pode ser útil no caso de existir alguma avaria no modo automático;
- Controlar o elevador em modo automático. Para o modo automático foram definidas 3 diferentes sequências de funcionamento do elevador. Na sequência 1, a cabina recebe uma ordem para subir (%Q2.1), desloca-se do piso 0 para piso 1 e, após 4 segundos, recebe uma ordem para descer para o piso 0 (%Q2.0). Na sequência 2, o funcionamento é semelhante, a cabina vai do piso 0 para o piso 2 e volta ao piso 0. Na sequência 3, a cabina vai para o piso 2, começa a descer parando no piso 1 e por fim no piso 0;
- Foram utilizadas algumas características do protótipo, como o sensor AF e os fins de curso FCI e FCS, para o controlo do elevador;
- O código em linguagem IL está associado às sequências de funcionamento em modo automático. É possível ir acompanhando pelo código em IL a sequência de funcionamento enquanto o elevador se movimenta;
- Ao botão P1, do modo manual, e ao botão PE também está associado código em linguagem IL;
- O botão de paragem de emergência (PE) imobiliza a cabina, em caso de alguma avaria.

É importante ainda referir que o simulador será útil no ensino das linguagens de programação lista de instruções (IL), linguagem *Ladder* (LD) e linguagem SFC. O simulador poderá ainda ser utilizado como recurso didático no ensino à distância.

5.2 Trabalho Futuro

Um dos objectivos principais no desenvolvimento do simulador de elevador do tipo monta-cargas é a sua utilização no ensino académico, como recurso didático no ensino das linguagens de programação lista de instruções (IL), linguagem *Ladder* (LD) e linguagem SFC, com a possibilidade de vir a ser utilizado para realizar testes e simulações, para posterior implementação em autómatos e *kits* disponíveis em laboratório.

Os principais pontos de estudo a desenvolver num trabalho futuro são:

- Trabalhar na otimização e robustez do simulador, melhorando o seu desempenho, comportamento e aspeto gráfico;
- Adicionar ao simulador outros tipos de sensores de posição, de sinal analógico, como os sensores de ultrassom, permitindo obter a posição exata do elevador;
- Introduzir um modelo físico mais realista, descrevendo o comportamento do sistema através de equações diferenciais, incluindo a parte da cinemática e dinâmica do modelo;
- Modular o comportamento do variador de velocidade (VF);
- Recurso a outras linguagens de programação para interpretação dos modos de funcionamento do elevador;
- Converter a parte gráfica do elevador de 2D para 3D;
- Construção de um compilador de linguagens de programação baseado na tecnologia de *Lex* e *Yacc*.

BIBLIOGRAFIA

- [1] Autostadt. *Car Towers*. (Consult. Março 2020). URL: <https://www.autostadt.de/explore/car-towers>.
- [2] H. Barragán. *Wiring*. (Consult. Março 2020). URL: <http://wiring.org.co/>.
- [3] J. K. Debnath e G. Serpen. “Real-time optimal scheduling of a group of elevators in a multi-story robotic fully-automated parking structure”. Em: *Procedia Computer Science* 61 (2015), pp. 507–514.
- [4] *Decreto-Lei 320/2002, 2002-12-28*. URL: <https://dre.pt>.
- [5] *Decreto-Lei 513/70, 1970-10-30*. URL: <https://dre.pt>.
- [6] *Decreto-Lei 58/2017, 2017-06-09*. URL: <https://dre.pt>.
- [7] I. Greenberg. *Processing: Creative Coding and Computational Art*. Friends of, 2007.
- [8] IEC. *IEC 61131-3 Standard - Programmable controllers - Part 3: Programming languages*. Jan. de 2003.
- [9] K.-H. John e M. Tiegelskamp. “The New Programming Languages of IEC 61131-3”. Em: *IEC 61131-3: Programming Industrial Automation Systems* (2001).
- [10] A Maria. “Introduction to modelling and simulation”. Em: *Winter Simulation Conference* (1997), pp. 7 –13.
- [11] nebulaeandstars. *Jelly Sim*. (Consult. Março 2020). URL: <https://www.openprocessing.org/sketch/587065>.
- [12] NicholasN. *Create a Game with Arduino and Processing*. (Consult. Março 2020). URL: https://create.arduino.cc/projecthub/Nicholas_N/create-a-game-with-arduino-and-processing-0fbcd3?ref=user&ref_id=234234&offset=1.
- [13] *NP-EN-81-1-2000*. (Consult. Março 2020). URL: <https://pt.scribd.com/document/350619562/NP-EN-81-1-2000>.
- [14] L. Palma. *Programação do Autómato TSX3721 - Introdução*. Universidade Nova de Lisboa - FCT - DEE, 2013.
- [15] L. Palma. *Automação Industrial - Programação dos Autómatos TSX3721 e Twido - Introdução*. Universidade Nova de Lisboa - FCT - DEE, 2014.
- [16] L. Palma. *Automação Industrial - Programação do Autómato TSX 3721 - Linguagem LD*. Universidade Nova de Lisboa - FCT - DEE, 2014a.

- [17] L. Palma. *Automação Industrial - Programação do Autómato TSX 3721 - Linguagem FBD*. Universidade Nova de Lisboa - FCT - DEE, 2014b.
- [18] L. Palma. *Automação Industrial - Programação do Autómato TSX 3721 - Linguagem SFC*. Universidade Nova de Lisboa - FCT - DEE, 2014c.
- [19] L. Palma. *Automação Industrial - Programação do Autómato TSX 3721 - Linguagem IL*. Universidade Nova de Lisboa - FCT - DEE, 2014d.
- [20] L. Palma. *Automação Industrial - Programação do Autómato TSX 3721 - Linguagem ST*. Universidade Nova de Lisboa - FCT - DEE, 2014e.
- [21] L. Palma. *Automation Systems Design, Specifications and Processes*. Universidade Nova de Lisboa - FCT - DEE, 2019.
- [22] PLCdev. *How PLCs Work*. (Consult. Março 2020). URL: http://www.plcdev.com/how_plcs_work.
- [23] E. E. Portal. *PLC handbook – Practical guide to programmable logic controllers*. AUTOMATIONDIRECT, 2017.
- [24] Processing. *lerp()*. (Consult. Março 2020). URL: https://processing.org/reference/lerp_.html.
- [25] C. Reas e B. Fry. *Processing: A Programming Handbook for Visual Designers and Artists*. Massachusetts Institute of Technology, 2007.
- [26] C. Reas e B. Fry. *Getting Started with Processing: A Hands-On Introduction to Making Interactive Graphics*. Maker Media, Inc., 2015.
- [27] C. Reas e B. Fry. *Processing*. (Consult. Março 2020). URL: <https://processing.org/>.
- [28] J. Rosas, L. Palma e P. Gil. “Virtual Environment for Simulation of Automation and Control Algorithms”. Em: *15th IEEE China-Europe International Symposium on Software Engineering Education (CEISEE), May 30-31, Caparica – Lisboa - Portugal* (2019).
- [29] A. Schlegel. *controlP5*. (Consult. Março 2020). URL: <http://www.sojamo.de/libraries/controlP5>.
- [30] Schneider. *Modicon M221*. (Consult. Março 2020). URL: <http://se.com/pt/pt/product-range/62128-controlador-1>.
- [31] Schneider. *Modicon M340*. (Consult. Março 2020). URL: <http://se.com/pt/pt/product-range/1468-modicon-m340/>.
- [32] S. Serban. *fractalLand*. (Consult. Março 2020). URL: <https://www.openprocessing.org/sketch/35608>.
- [33] E. Winsberg. “Computer Simulations in Science”. Em: *The Stanford Encyclopedia of Philosophy*. Ed. por E. N. Zalta. Winter 2019. Metaphysics Research Lab, Stanford University, 2019.

- [34] P. Zhang. *Advanced Industrial Control Technology*. William Andrew, 2010.
- [35] N. Ângelo. *Projecto e Concepção de um Sistema Elevador Monta-cargas Industrial*. Tese de mestrado. Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, 2016.



APÊNDICE

A.1 *Processing* - Função *Lerp()*

No desenvolvimento do simulador, em linguagem *Processing*, foi utilizada a função *lerp()* [24].

- **Sintaxe:** `lerp(start, stop, easing)`
- **Parâmetros:**
 - **start** – float: primeiro valor;
 - **stop** – float: segundo valor;
 - **easing** – float: entre 0.0 – 1.0

Esta função é importante para o programa pois permite movimentar o elevador entre os diferentes pisos - P0, P1 e P2.

A função utilizada no programa foi a seguinte:

```
y = lerp(y, target, easing)
```

Figura A.1: Função *Lerp()* - Movimento do Elevador.

Em que:

- **y** — Posição inicial do elevador (pisso atual);
- **target** — Posição final (pisso destino);
- **easing** — Incremento de posição (1%).



ANEXOS

I.1 Anexo 1 - Entradas e Saídas Disponíveis no PLC

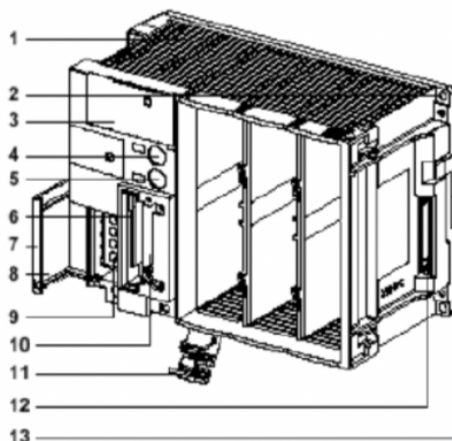
Na tabela [I.1](#) são apresentadas as entradas e saídas disponíveis no PLC, retirado de [\[35\]](#).

Tabela I.1: Lista de endereços de ligação das entradas/saídas do PLC TSX Micro 3721

Descrição	Endereço no PLC	Tipo de Sinal	Tipo de Contacto
Botão B0 ou Reset	%I1.0	ED	NA
Botão B1	%I1.1	ED	NA
Botão B2	%I1.2	ED	NA
Botão PE	%I1.3	ED	NF
Fim de Curso Superior (FCS)	%I1.4	ED	NF
Fim de Curso Inferior (FCI)	%I1.5	ED	NF
Sensor AF	%I1.6	ED	NA
Sensor AT	%I1.7	ED	NA
Sensor BF	%I1.8	ED	Memória
Sensor BT	%I1.9	ED	Memória
Relé Descida (KM0)	%Q2.0	SD	–
Relé Subida (KM1)	%Q2.1	SD	–
Relé Travão (KF2)	%Q2.2	SD	–
Relé Travão (KF3)	%Q2.3	SD	–
Sinal do Variador de Frequência	%IW5.0	EA	–
Sinal do PLC	%QW5.4	SA	–

I.2 Anexo 2 - Características PLC TSX Micro 3721

A Figura I.1 mostra os pontos característicos do PLC TSX Micro 3721 com a respectiva legenda.



- | | |
|---|---|
| 1 - Autómato (PLC) | 8 – Etiqueta |
| 2 - Ponto de montagem | 9 - Terminais de alimentação |
| 3 - Visor (“display”) | 10 - Ranhura para placa de comunicação Modbus (tipo PCMCIA) |
| 4 - Terminal TER | 11 - Tampa da bateria |
| 5 - Terminal AUX | 12 - Conector para módulo de extensão |
| 6 - Ranhura para extensão de memória | 13 - Pontos de montagem DIN |
| 7 - Tampa dos terminais da fonte de alimentação | |

Figura I.1: PLC TSX Micro 3721 [14, 35].

No PLC TSX Micro 3721 (base com extensão), conforme apresentada na Figura I.2, cada módulo de entradas/saídas (*Input/Output (I/O)*) pode ocupar as posições 1 e 10.

A posição 0 está reservada para a CPU e para módulos integrados.

O endereçamento dos canais depende da localização geográfica do módulo no PLC.

A Figura I.2 mostra a localização do módulo DMZ 28DR nas posições 1 e 2, na qual as entradas estão situadas na posição 1 e as saídas estão situadas na posição 2.

A configuração da estrutura de endereçamento de entradas/saídas serve para atribuição e caracterização das variáveis utilizadas nas linguagens de programação do PLC TSX Micro 3721.

Por exemplo, as entradas estão no módulo 1 (posição 1 do *rack*) e são referências por:

- %I1.x, x =0, 1, 2, ..., 15, em que x corresponde à entrada do módulo;
- %I1.2 significa a entrada 2 do módulo 1.

As saídas estão no módulo 2 (posição 2 do *rack*) e são referências por:

- %Q2.x, x =0, 1, 2, ..., 11, em que x corresponde à saída do módulo;
- %Q2.1 significa a saída 1 do módulo 2.

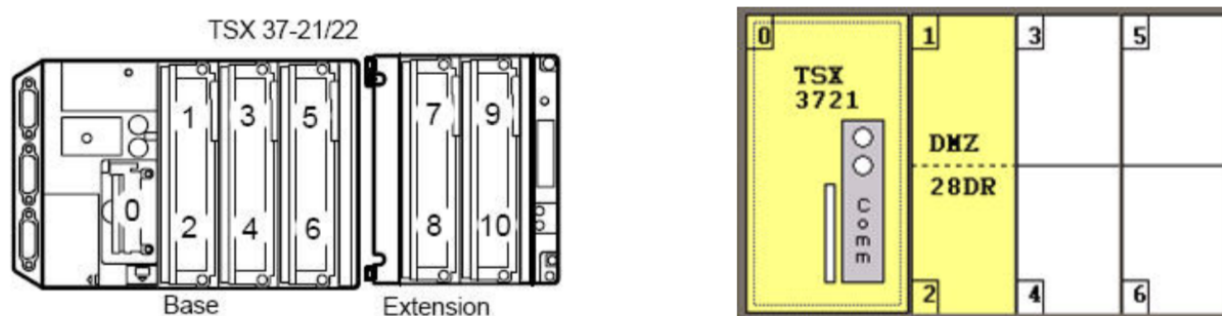


Figura I.2: PLC base e extensão e localização do módulo DMZ 28DR [14, 35].

Na Figura I.3 é apresentada as ligações do PLC TSX Micro 3721.

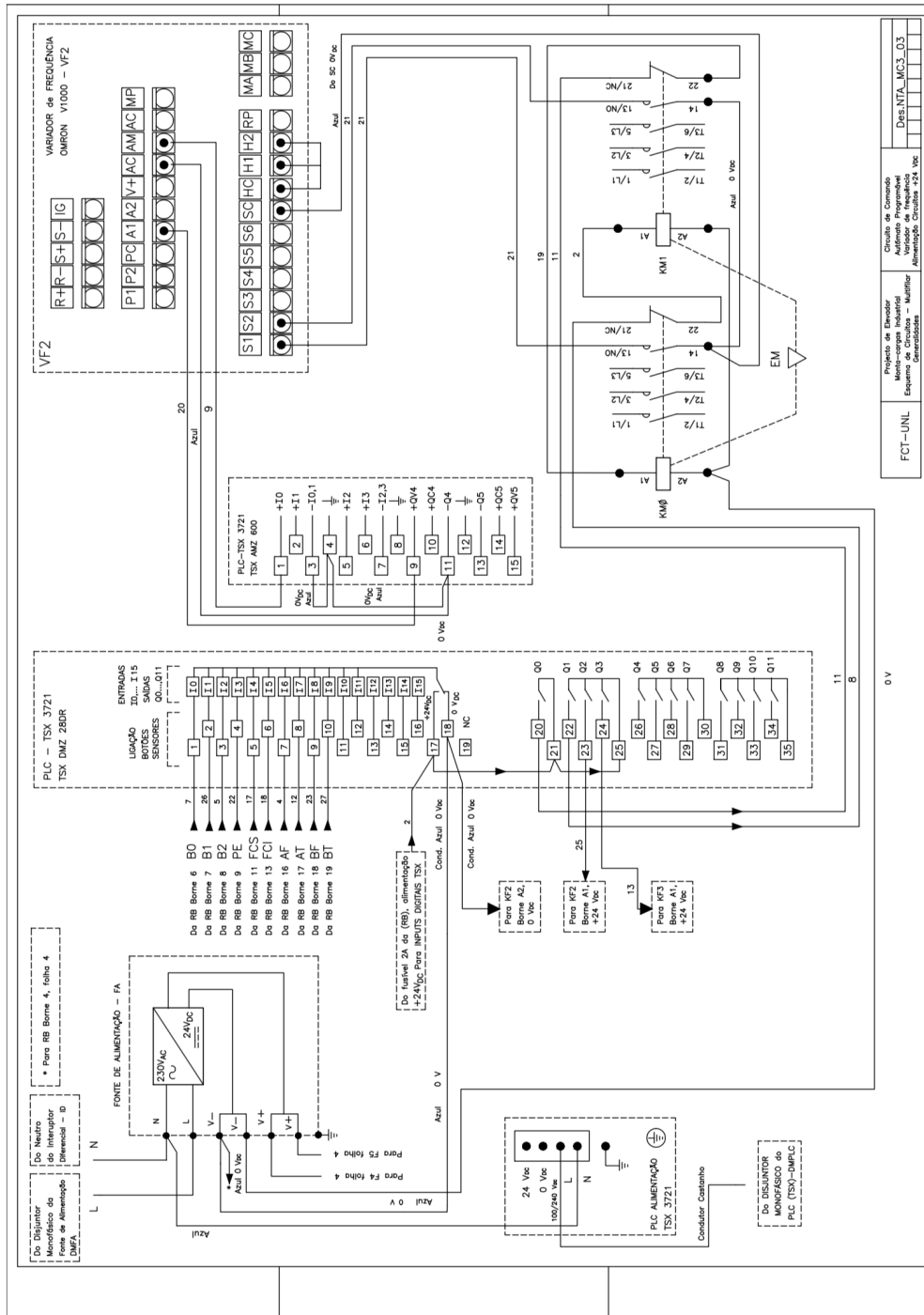


Figura I.3: Ligações do PLC TSX Micro 3721 [35].

I.3 Anexo 3 - Disposição dos Sensores, dos Fins de Curso e dos Ímanes

Na Figura [I.4](#) é apresentada a disposição dos sensores, dos fins de curso e dos ímanes ao longo das guias da estrutura do protótipo de elevador monta-cargas, retirado de [\[35\]](#).

I.3. ANEXO 3 - DISPOSIÇÃO DOS SENSORES, DOS FINS DE CURSO E DOS ÍMANES

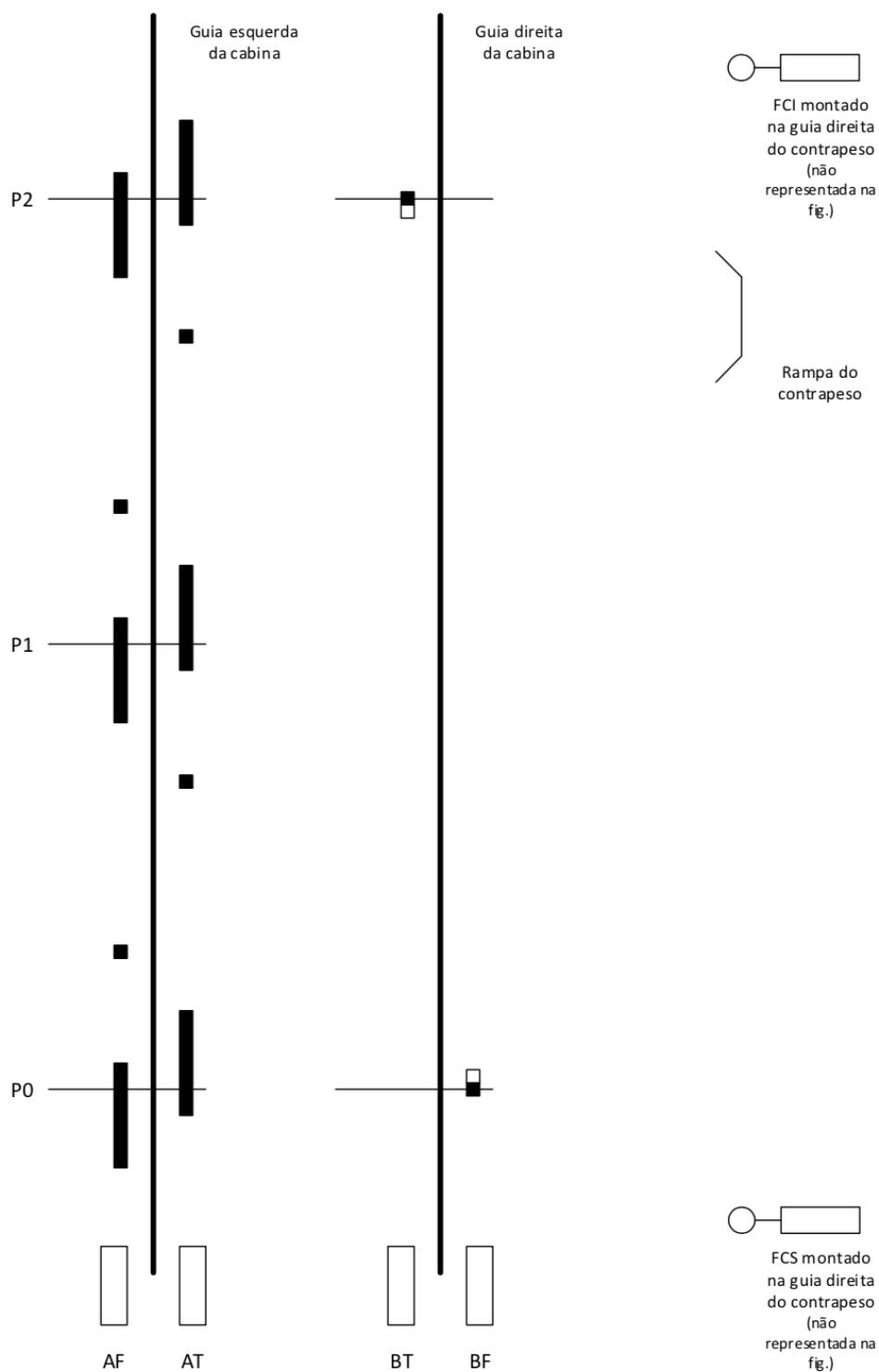


Figura I.4: Disposição dos sensores, dos fins de curso e dos ímanes ao longo das guias [35].